

Location noticer

App para notificar cuando alguien sale de una zona delimitada en un mapa como segura



Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:
Ángel López Carrión

Tutor/es:
José Vicente Berná Martínez

Julio 2021



Universitat d'Alacant
Universidad de Alicante

Resumen

Actualmente, en el mundo hay un gran número de personas que son dependientes y necesitan de la supervisión o vigilancia de otras. En su gran mayoría, estas personas suelen ser menores, pertenecientes al grupo de la tercera edad o personas con alguna enfermedad o discapacidad.

No es raro encontrar en redes sociales y barrios anuncios de personas dependientes e indefensas que se han perdido o desaparecido y se pide ayuda para encontrarlos.

Por su seguridad, deben tener a alguien que les tenga a su cuidado, aunque en ocasiones es inevitable que esas personas estén solas. Como todas las personas, necesitan su libertad, aunque en muchas ocasiones tampoco se puede controlar su vida al 100% y cambiarla.

Dado el evidente crecimiento del uso de las nuevas tecnologías en todas las edades y ámbitos, se puede proponer una solución a estos problemas. Una alternativa a otras soluciones del mercado que apuestan por seguros, vigilancia 24h o servicios que se deben contratar conjuntos.

Location noticer nace de esa necesidad, para aportar seguridad a una persona que está al cargo de otra, de poder saber que está en un sitio seguro y, en caso de no estarlo, saberlo de manera inmediata e incluso que te indique donde se encuentra.

Además, esta aplicación no es secreta, buscando evitar posibles espionajes o violaciones de la privacidad de una persona, la aplicación no se encuentra oculta y la persona vigilada ha permitido que le sigan desde la propia app. En caso contrario, la misma persona vigilada podría eliminar a la persona que quiera que le esté realizando el seguimiento.

Por lo tanto, el objetivo principal de este proyecto es, haciendo uso de la aplicación de Location noticer, notificar mediante una alarma cuando una persona se encuentra fuera de una zona delimitada como segura en un mapa.

Resum

Actualment, al món hi ha un gran nombre de persones que són dependents i necessiten de la supervisió o vigilància d'altres. La gran majoria d'aquestes persones solen ser menors, tindre una edat avançada o són persones amb alguna malaltia o discapacitat.

No és estrany trobar a les xarxes socials i barris anuncis de persones dependents i indefenses que s'han perdut o desaparegut i es necessita ajuda per trobar-los.

Per la seva seguretat, deuen tenir a algú que els tinga al seu càrrec, encara que en ocasions és inevitable que aquestes persones estiguen a soles. Com totes les persones, necessiten la seua llibertat, encara que en moltes ocasions tampoc es pot controlar la seua vida al 100% i canviar-la.

Donat l'evident creixement de l'ús de les noves tecnologies en totes les edats i àmbits, es pot proposar una solució a aquests problemes. Una alternativa a altres solucions del mercat que aposten per segurs, vigilància 24h o servicis que es deuen contractar conjunts.

Location noticer naix d'aquesta necessitat, per aportar seguretat a una persona que està al carrer d'altra, de poder conèixer que està a un lloc segur i, en cas de no estar, conèixer-ho de forma immediata i que t'indique on es troba.

A més a més, aquesta aplicació no és secreta, buscant evitar possibles espionatjes o violacions de la privacitat d'una persona, l'aplicació no es troba oculta i la persona vigilada ha permés que li seguisquen a la propia aplicació. Al cas contrari, la mateixa persona vigilada podria esborrar a la persona que vulga que estiga fent el seguiment.

Per tant, l'objectiu principal d'aquest projecte és, amb l'aplicació de Location noticer, notificar mitjançant una alarma quan una persona es troba fora d'una zona delimitada com segura a un mapa.

Motivación, justificación y objetivo general

Cuando piensas en la elección de tema o propuesta para este trabajo fin de grado te planteas inicialmente ideas que te gustaría hacer como proyectos personales o simplemente por gusto. Tenía claro que quería hacer una propuesta mía y así ha sido. Propuse una idea base que sería la geolocalización en una aplicación.

La idea surgió debido a que tengo un familiar cercano, un abuelo, con principio de Alzheimer, el cual cada día necesita más nuestra atención y ayuda. Cuando se me ocurrió la idea, él todavía tenía una gran independencia, lo que me hizo pensar que para familiares sería muy tranquilizador tener una especie de localizador que les avisase cuando se alejase de las zonas que suele habitar. Finalmente, partiendo de esa idea y del refinamiento de ésta junto a mi tutor José Vicente Berná, concretamos que el tema que no sería exclusivamente para localizar personas con Alzheimer, ya que se podría necesitar en más casos, y que para hacer la localización la haríamos con dos aplicaciones en distintos smartphones, en lugar de un localizador GPS.

Existen gran cantidad de aplicaciones para consultar la localización de personas mayores o con necesidades especiales, muchas de ellas incluso las ofertan empresas de seguros. Lo que busco diferencialmente con mi idea es que ésta se base en la notificación a los familiares de las salidas de unas zonas comunes remarcadas, no de un seguimiento paso a paso del recorrido o dónde se encuentre la persona en cada momento, aunque esas opciones no estarían descartadas inicialmente.

Además, mi objetivo con el TFG también es aprender independientemente a hacer un proyecto de este calibre. Con el proyecto ABP de 4º de Ingeniería Multimedia aprendí a trabajar en grupo, aunque me quedé con una sensación de que podría haber aprendido más y haber mejorado individualmente que me gustaría subsanar en este proyecto.

Agradecimientos

En primer lugar, quiero agradecer a mi familia, a mis padres, mis abuelos y mis tíos, y a Sara por todo el apoyo, confianza y fe ciega en mí. Espero habérsela devuelto, gracias de verdad.

Gracias también a José Vicente por el apoyo y ayuda en la realización de este proyecto como hace un año durante el ABP. Has sido un gran profesor, tutor y profesional al cual pedir consejo y aprender continuamente. También gracias al resto de profesionales y docentes de la universidad y la carrera que me han ayudado a aprender y a crecer en conocimientos y profesionalmente.

Gracias a todos mis amigos que todavía seguís ahí, por aguantarme y apoyarme. Sois los mejores, de verdad, gracias.

Gracias a la comunidad de Multitec, con vosotros he crecido y sois un gran ejemplo a seguir y apoyo en casi todo. Ha sido una gran experiencia estar con vosotros y espero ser para vosotros lo que sois vosotros para mí.

Gracias a todos de verdad, este trabajo va para vosotros.

Citas

*Nadie consigue nada por suerte. Si trabajas duro, seguro que tendrás
suerte. La suerte se trabaja.*

Lola Carrión, mi madre

Do it Better.

Comunidad de estudiantes MultitecUA

Índice de contenidos

Resumen.....	2
Resum.....	3
Motivación, justificación y objetivo general	4
Agradecimientos	5
Citas.....	6
Índice de figuras	10
Índice de tablas	13
1. Introducción	14
2. Estudio de viabilidad	16
2.1. Análisis DAFO	16
2.2. Lean Canvas.....	17
2.3. Análisis de riesgos	20
3. Planificación	25
4. Estado del arte	27
4.1. Localización en móviles	27
4.1.1. Sistema A-GPS	28
4.1.2. Posicionamiento basado en WiFi	28
4.1.3. Por direcciones IP	29
4.1.4. Geolocalización GSM	30
4.2. Ejemplos de utilización de la localización	30
4.2.1. En vehículos de transporte de mercancías	30
4.2.2. Antirrobo	32
4.2.3. En aves y otros animales	33
4.2.4. En pulseras para presos.....	34
4.3. Soluciones existentes	34
4.3.1. LoPe (Localizador de personas).....	35

4.3.2.	Weenect	36
4.3.3.	Find My Kids	37
4.3.4.	Life360	38
4.3.5.	GeoZilla.....	39
4.3.6.	AlertCops	40
4.3.7.	Conclusiones.....	41
5.	Objetivos	43
6.	Metodología	45
7.	Análisis y especificación	48
7.1.	Tipos de usuarios.....	48
7.2.	Requisitos	49
7.2.1.	Requisitos funcionales.....	49
7.2.2.	Requisitos no funcionales	50
8.	Diseño.....	53
8.1.	Diseño arquitectura conceptual.....	53
8.2.	Diseño de la arquitectura tecnológica	54
8.3.	Diseño de la persistencia.....	55
8.3.1.	Almacenamiento de datos	55
8.3.2.	Avance en el diseño del modelo de datos.....	58
8.3.3.	Seguridad e integridad	60
8.4.	Diseño Interacción o Experiencia de Usuario.....	61
8.4.1.	Tipos de personas e intereses	61
8.4.2.	User Journey Map	62
8.5.	Guía de estilos	65
8.5.1.	Logotipo e icono	65
8.5.2.	Colores corporativos	66
8.5.3.	Fuente y tipografía	67
8.6.	Diseño Interfaces.....	68

8.7.	Avance en el diseño de interfaces.....	75
8.8.	Diseño de pruebas y validación.....	82
9.	Implementación	84
9.1.	Sprint 1: Preparación y configuración del proyecto.....	84
9.2.	Sprint 2: Primeras interfaces y la gestión de usuarios	87
9.3.	Sprint 3: Enlace entre dispositivos a través del código.....	91
9.4.	Sprint 4: Inclusión del mapa.....	97
9.5.	Sprint 5: Seguimiento con ubicación real.....	100
9.6.	Sprint 6: Alarma y configuración	104
9.7.	Sprint 7: Últimos detalles y corrección de errores.....	108
10.	Pruebas y validación.....	112
11.	Resultados	115
11.1.	Producto final	115
11.2.	Costes temporales.....	120
11.3.	Asignaturas relacionadas	122
12.	Conclusiones y trabajo futuro	123
12.1.	Comprobación de objetivos	123
12.2.	Trabajo futuro y mejoras.....	123
12.3.	Impresiones personales	125
	Referencias.....	127

Índice de figuras

Figura 1. Análisis DAFO.....	16
Figura 2. Lean Canvas.....	20
Figura 3. Esquema del funcionamiento del sistema A-GPS.....	28
Figura 4. Ejemplo de geolocalización por IP online.....	29
Figura 5. Interfaces de aplicación web para seguimiento, monitoreo y control de maquinaria pesada	32
Figura 6. Interfaces de Weenect	36
Figura 7. Interfaces de Find My Kids	38
Figura 8. Interfaces de Life360	39
Figura 9. Interfaces de GeoZilla.....	40
Figura 10. Interfaces de AlertCops y Guardián	41
Figura 11. Captura del tablero Kanban del proyecto en Trello	46
Figura 12. Captura del estado actual de la herramienta Toggl	47
Figura 13. Diseño Arquitectura Conceptual	53
Figura 14. Esquema Arquitectura Tecnológica.....	54
Figura 15. Modelo de datos	57
Figura 16. Actualización del modelo de datos	59
Figura 17. User Journey Map	63
Figura 18. Icono Location noticer.....	65
Figura 19. Logotipo Location noticer.....	66
Figura 20. Colores corporativos	66
Figura 21. Fuente KoHo	67
Figura 22. Wireframes de Location Noticer	69
Figura 23. Todos los mockups desde Figma	70
Figura 24. Interfaces 1, 2 y 3	71
Figura 25. Interfaces 4, 5, 6 y 7	72
Figura 26. Interfaces 8 y 9	73
Figura 27. Interfaces 10 y 11	74
Figura 28. Mapa de interacción de interfaces.....	75
Figura 29. Mockups finales.....	76

Figura 30. Interfaces 1, 2 y 3	77
Figura 31. Interfaz 4	78
Figura 32. Interfaces 5, 6 y 7	79
Figura 33. Interfaces 8 y 9	80
Figura 34. Interfaces 10 y 11	81
Figura 35. Directorio del proyecto de Ionic en Visual Studio Code.....	85
Figura 36. Firestore Database configurado con ejemplos introducidos	86
Figura 37. Firebase Authentication habilitada con email y contraseña.....	87
Figura 38. Comando para generar una nueva página desde terminal en Ionic	88
Figura 39. Vista de las nuevas páginas en el directorio	88
Figura 40. Guardado del usuario del registro en Firebase Authentication y Firestore Database	89
Figura 41. Interfaces 1, 2 y 3 implementadas	91
Figura 42. Ejemplo del uso de iconos de Fontello.....	91
Figura 43. Iconos descargados e incluidos en el proyecto	92
Figura 44. Ejemplo de mensaje modal en Location Noticer	93
Figura 45. Visualización de la página generar-codigo	94
Figura 46. Consulta y borrado de colecciones y documentos en Firestore Database	95
Figura 47. Visualización de la página introducir-codigo.....	96
Figura 48. Inicialización del mapa con Leaflet y creación de un circle.....	98
Figura 49. Visualización de la página mapa con estado de selección de rango	99
Figura 50. Visualización de la página mapa con estado de seguimiento	101
Figura 51. Captura de la aplicación desde el menú de un teléfono Android	103
Figura 52. Capturas de las interfaces de la APK vistas desde un Samsung Galaxy A50	103
Figura 53. Visualización de la página de alarma	105
Figura 54. Visualización de la página mapa en estado de alarma.....	106
Figura 55. Visualización de la interfaz config para un usuario Protegido y para un Guardián .	107
Figura 56. Visualización de un mensaje modal de confirmación en la página config.....	108
Figura 57. Documento de la colección Seguimiento tras generar un nuevo código de enlace	109
Figura 58. Ejemplo nueva distancia radial máxima en la selección de rango	111
Figura 59. Panel de tiempos de respuesta de Firebase Performance Monitoring	113
Figura 60. Panel de control de Firebase Performance Monitoring.....	113
Figura 61. Primeras interfaces para un usuario Protegido de la demostración	115
Figura 62. Primeras interfaces para un usuario Guardián de la demostración	117
Figura 63. Interfaces de configuración del seguimiento para un usuario Guardián de la demostración	118

Figura 64. Interfaces de alarma para un usuario Guardián de la demostración.....	119
Figura 65. Horas invertidas en el TFG por mes	120
Figura 66. Horas dedicadas a cada apartado de la memoria	121

Índice de tablas

Tabla 1. Identificación y clasificación de riesgos.....	21
Tabla 2. Análisis de estrategias con respecto a los posibles riesgos.....	24
Tabla 3. Planificación temporal TFG.....	25
Tabla 4. Tipos de usuario en la aplicación.....	48
Tabla 5. Requisitos funcionales.....	50
Tabla 6. Requisitos no funcionales.....	52

1. Introducción

Son muchos los casos en los que hoy en día se necesita una ayuda tecnológica para nuestros problemas del día a día. Hay gran cantidad de casos en los que puede ser necesario controlar la ubicación de una persona, animal u objeto.

Hay una larga lista de personas, ya sean menores, ancianos o personas con alguna afección o enfermedad, que pueden necesitar ser localizados en algún momento para que sus familiares estén más tranquilos.

Normalmente, estos son casos que no llegan a ser sobre personas totalmente dependientes ni totalmente independientes, personas en principio independientes pero que necesitan de un mínimo control y supervisión.

Para casos de ancianos, hay gran cantidad de seguros que incluyen localizadores para personas con enfermedades degenerativas. Y en el mercado también hay cantidad de aplicaciones para localizar smartphones, pero en su gran mayoría debes entrar a ella para comprobar la posición de estos.

Para los casos en los que se necesita saber inmediatamente si una persona ha salido de una zona que se puede considerar segura, no hay aplicaciones que te aporten ese tipo de flexibilidad e instantaneidad.

Para aquellos que están al cargo y supervisando a estas personas, sería de gran ayuda una aplicación sencilla e intuitiva que les avisase cuando éstas salieran de sus zonas habituales y seguras.

Debido a estos motivos, decidí en su día que mi Trabajo de Fin de Grado tratase sobre este tema, desarrollar una herramienta que ayude a notificar y conocer cuando una persona sale de su zona habitual de tránsito.

La aplicación principal, que será la que administren las personas que localizar a otras, podrán decidir cuantas y cuáles serán las zonas seguras del localizador, además de poder consultar recorrido cuando se sale de estas zonas, un historial para posibles consultas y otras tantas funcionalidades que puedan ser necesarias para la mejor atención de la persona y buen uso de la información que se aporta.

Esta aplicación no pretende ser una herramienta para espiar o reducir las libertades de nadie, busca ayudar a personas, está dirigida a esos ámbitos y pretende que siempre sea con consentimiento y conocimiento mutuo entre localizador y localizado.

En este documento, se profundizará en los detalles del desarrollo de esta app móvil en todos sus pasos, desde el refinamiento de la idea y el diseño hasta el testing y final del desarrollo.

2. Estudio de viabilidad

En todo proyecto, entre los primeros pasos siempre se encuentran estudios de viabilidad. Es una forma de analizar el proyecto, sus puntos fuertes y débiles y así poder hacer una mejor toma de decisiones sobre la idea o partes de su desarrollo.

Un buen estudio de viabilidad es fundamental para confirmar o refinar la idea inicial y poder mejorar de cara al desarrollo y futuro producto final.

Este apartado mostrará un análisis DAFO, un Lean Canvas y un análisis de riesgos y el desarrollo de estos.

2.1. Análisis DAFO

El análisis DAFO también conocido como análisis FODA, es una metodología de estudio de la situación de una empresa o un proyecto, analizando sus características internas (Debilidades y Fortalezas) y su situación externa (Amenazas y Oportunidades). De ahí, el nombre que adquiere. Proviene de las siglas en inglés SWOT (Strengths, Weaknesses, Opportunities y Threats) [1]. Se muestra en la Figura 1 como ejemplo visual el esquema del análisis DAFO que desarrollo a continuación.

	De origen interno	De origen externo
Puntos débiles	Debilidades Proyecto individual. Falta de experiencia. Desconocimiento de herramientas y uso de información. Baja familiaridad del usuario con smartphones. Servicio poco sostenible en el tiempo.	Amenazas Competidores con mayor conocimiento del sector. Otras apps ofrecen servicios similares.
Puntos fuertes	Fortalezas Su uso no implica la contratación de otros servicios. Las notificaciones son prácticamente instantáneas.	Oportunidades Los mayores competidores no ofrecen exactamente el mismo servicio. Otros no ofrecen seguridad con el uso de los datos obtenidos.

*Figura 1. Análisis DAFO.
(Fuente propia)*

Este sería en análisis DAFO correspondiente al proyecto:

- Debilidades:
 - Primer gran proyecto individual.
 - Falta de experiencia desarrollando y manteniendo una aplicación.
 - Desconocimiento de las herramientas correctas a usar.
 - Desconocimiento del correcto uso de la información que puede contener u ofrecer la herramienta.
 - En ocasiones, a quien le pueda hacer falta no es una persona que acostumbre a usar o haya usado alguna vez un smartphone.
 - El servicio mínimo gratuito es el necesario y esencial para el correcto funcionamiento y cubrir la necesidad principal; los extras de pago no serían necesarios, simplemente complementarios. Esto hará al servicio que no sea sostenible en el tiempo.
- Amenazas:
 - Los competidores tienen un mayor conocimiento del sector.
 - Otras aplicaciones ofrecen servicios similares.
- Fortalezas:
 - Su descarga y uso no implica la contratación de otros servicios, como podrían ser seguros o localizadores.
 - La herramienta notifica prácticamente instantáneamente las salidas
- Oportunidades:
 - Los mayores competidores no ofrecen exactamente el mismo servicio.
 - Otros no ofrecen la seguridad del buen uso y privacidad de los datos obtenidos.

2.2. Lean Canvas

El Lean Canvas es una herramienta estratégica empresarial, que permite analizar de manera visual nuestro modelo de negocio para aumentar sus probabilidades de éxito.

Este se basa en 9 pilares: la definición de los clientes objetivos, identificar los problemas que tiene el cliente y que el producto puede solucionar, una proposición de valor única donde se indica lo que ofrece el producto para solucionar estos problemas, listar las características principales que solucionarán el problema, identificar los canales que lo darán a conocer, definir las vías de ingresos, analizar los principales costes, establecer las métricas clave que nos indiquen si se está desarrollando correctamente el modelo de negocio y expresar aquello que hace al producto especial frente a la competencia. [2]

A continuación, se van a desarrollar más a fondo estos puntos en lo referido al proyecto:

- Segmentos de clientes:

Encargados o supervisores de personas dependientes, ancianos o menores en la edad de la adolescencia.

Los usuarios visionarios (o early adopters) del producto, que son los primeros que apostarían por el producto, serían encargados de cuidar a personas mayores que inicialmente son independientes (aunque cada vez más dependientes) y viven solos.

- Problema:

Dejar solas a personas dependientes y la inseguridad que provoca en sus familiares

En la actualidad, la alternativa al producto para cómo solventarlo, sería el uso de sistemas de geolocalización y la consulta de forma repetitiva de su estado.

- Proposición de valor única:

Una aplicación que te mantendrá informado cuando una persona dependiente sale de su zona conocida o segura.

- Solución:

- Puedes no estar en constante alerta o vigilancia cuando no estás con ellos.
- Puedes tener una rápida reacción cuando la aplicación te avisa.
- Es una aplicación que siempre irá contigo activada.
- Te muestra el recorrido que hacen si se salen de la zona segura para poder encontrarlos o saber qué hacer rápidamente.

- Canales:

Mediante asociaciones y comunidades, anuncios en páginas web y, como canal de bajo impacto, de boca a boca.

- Flujo de ingresos:

En principio, al ser un trabajo de ámbito universitario, no se buscará tener un flujo de ingresos. Aunque encaja perfectamente en un modelo Freemium, donde se podrían añadir nuevas funcionalidades que tengan un coste para el usuario.

- Estructura de costes:

El coste actualmente sería el de recursos humanos y la publicación en tiendas de aplicaciones.

El precio total para publicar una app en la tienda Google Play Store será los 25 dólares (poco más de 20 euros) por darse de alta como desarrollador, que sólo se deberá abonar la primera vez y después el 30% de la facturación total de la venta de las aplicaciones que se tengan disponibles en la tienda también será para Google. [3]

La tarifa anual como desarrollador iOS de Apple (Apple Developer Program) es de 99 dólares al año. [4]

En un futuro podrían incrementarse si se ampliase a localizar también localizadores, puesto que habría que comprarlos y trabajar con la tecnología para recibir correctamente la señal de estos, así que también podría incluir el coste de un hosting y dominio para una API.

- Métricas clave:

- Números de descargas de la aplicación en las tiendas de aplicaciones.
- Cuántos usuarios hacen uso real de ésta.
- Funcionalidades más usadas.
- Qué público acoge mejor el servicio entre el público objetivo.
- Valoración en tiendas de aplicaciones.

- Ventaja especial:

La aplicación te ofrece avisos gratuitos instantáneos a un público más amplio.

Problema: Dejar solas a personas dependientes y la inseguridad que provoca en sus familiares. Alternativas actuales: Uso de sistemas de geolocalización y la consulta de forma repetitiva de su estado.	Solución: ·No estar en constante alerta o vigilancia. ·Poder tener una rápida reacción. ·Siempre irá contigo activada. ·Te muestra el recorrido que hacen si se salen de la zona segura. Métricas clave: ·Número de descargas. ·Usuarios que hacen uso real. ·Funcionalidades más usadas. ·Público que acoge mejor el servicio. ·Valoración en tiendas de aplicaciones.	Proposición de valor única: Una aplicación que te mantendrá informado cuando una persona dependiente sale de su zona conocida o segura.	Ventaja especial: La aplicación te ofrece avisos gratuitos instantáneos a un público más amplio. Canales: Mediante asociaciones y comunidades, anuncios en páginas web y, como canal de bajo impacto, de boca a boca.	Segmentos de clientes Encargados o supervisores de personas dependientes, ancianos o menores en la edad de la adolescencia. Early adopters (usuarios visionarios): Serían encargados de cuidar a personas mayores que inicialmente son independientes (aunque cada vez más dependientes) y viven solos.
Estructura de costes: El coste actualmente sería el de recursos humanos y la publicación en tiendas de aplicaciones. En un futuro podrían incrementarse si se ampliase a localizar también localizadores, puesto que habría que comprarlos y trabajar con la tecnología para recibir correctamente la señal de estos, así que también podría incluir el coste de un hosting y dominio para una API.			Flujo de ingresos: En principio, al ser un trabajo de ámbito universitario, no se buscará tener un flujo de ingresos. Aunque encaja perfectamente en un modelo Freemium, donde se podrían añadir nuevas funcionalidades que tengan un coste para el usuario.	

*Figura 2. Lean Canvas.
(Fuente propia)*

2.3. Análisis de riesgos

A lo largo de un proyecto es adecuado que, antes de iniciar el desarrollo de este, se pueda tener en consideración riesgos que pueden ocurrir para poder tener una correcta propuesta de planes de contingencia para facilitar el éxito del trabajo fin de grado.

En la siguiente tabla, se mostrará la identificación de riesgos y la clasificación de estos respecto a su probabilidad de ocurrir y los efectos que tendrían sobre el proyecto y el desarrollo.

La probabilidad de que un posible riesgo ocurra durante el desarrollo puede ser Baja (que ocurra raramente, una vez en la vida o ninguna), Media (que pueda ocurrir de vez en cuando, una vez al año) y Alta (que sea habitual, una vez a la semana o al mes) y el efecto que pueda tener sobre el proyecto puede ser Catastrófico, Serio, Tolerable o Insignificante.

Tipo de riesgo	Posible riesgo	Probabilidad	Efecto
Tecnología	Fallo de internet.	Baja	Catastrófico
	Mal rendimiento del producto final.	Media	Tolerable
	Deterioro o avería de ordenadores.	Media	Serio

	Pérdida de archivos.	Baja	Catastrófico
	Pérdida total del código y copias de seguridad.	Baja	Serio
Personas	Causar baja por accidente, enfermedad u otro.	Media	Tolerable
	No poder continuar con el normal desarrollo del proyecto para poder finalizarlo a tiempo.	Baja	Catastrófico
	Abandono del proyecto.	Baja	Catastrófico
	Agobio o estrés en el proceso de desarrollo del TFG.	Alta	Serio
Organización	Mala planificación.	Alta	Serio
	Retraso en el desarrollo con respecto a la planificación.	Alta	Tolerable/Serio
Herramientas	Desconocimientos de las tecnologías a utilizar.	Alta	Tolerable
	Perder el acceso a los servicios externos de los que haga uso la aplicación.	Baja	Serio
Requerimientos	Requerimientos de última hora.	Alta	Tolerable
	Cambios en requerimientos.	Baja	Serio
Otros	Que se presente en el mercado una idea o proyecto alternativo para solucionar el mismo problema.	Media	Tolerable/Serio

Tabla 1. Identificación y clasificación de riesgos

Para establecer las estrategias para cada uno de los riesgos valorados en la anterior tabla, éstas se dividen en tres categorías: Prevención, para reducir la probabilidad de que ese riesgo se produzca; Minimización, para disminuir el impacto de ese riesgo y Plan de Contingencia, para prepararse ante la peor situación posible.

Tipo de riesgo	Posible riesgo	Estrategia
Tecnología	Fallo de internet.	Prevención: Trabajar lo máximo posible en local.
	Mal rendimiento del producto final.	Prevención: Revisión continua de las funcionalidades y mejorarlas.
	Deterioro o avería de los ordenadores.	Minimización: Tener disponible otra vía de trabajo u otros ordenadores cercanos.
	Pérdida de archivos.	Prevención: Realizar copias de seguridad.
		Minimización: Tratar de recuperar y restaurar el mayor número de archivos posibles.
		Plan de Contingencia: Tener siempre disponibles copias de seguridad y versiones anteriores del proyecto.
	Pérdida total del código y copias de seguridad	Prevención: Tener más copias de seguridad en varios dispositivos.
		Minimización: Usar una copia de seguridad anterior si la hubiese.
Personas	Causar baja por accidente, enfermedad u otro.	Prevención: Tener en cuenta en la planificación un tiempo margen de reacción.
		Plan de Contingencia: Entregar y defender en la siguiente convocatoria o curso siguiente.

	No poder continuar con el normal desarrollo del proyecto para poder finalizarlo a tiempo.	Prevención: Tener en cuenta en la planificación un mayor tiempo margen de reacción.
		Plan de Contingencia: Entregar y defender en la siguiente convocatoria o curso siguiente.
	Abandono de proyecto.	Prevención: Mantener el interés y realizar una correcta planificación.
	Agobio o estrés en el proceso de desarrollo del TFG.	Prevención: Realizar una correcta planificación.
Organización	Mala planificación.	Minimización: Realizar con prioridad las funcionalidades básicas y esenciales.
	Retraso en el desarrollo con respecto a la planificación.	Prevención: Hacer una planificación correcta con márgenes de tiempo y revisarla periódicamente.
		Minimización: Incrementar el tiempo de trabajo invertido y la productividad.
Herramientas	Desconocimiento de las tecnologías a utilizar.	Prevención: Hacer una planificación correcta con márgenes de tiempo y revisarla periódicamente.
		Minimización: Incrementar el tiempo de trabajo invertido y la productividad.
Herramientas	Desconocimiento de las tecnologías a utilizar.	Prevención: Aprender y estudiar las tecnologías a utilizar de forma previa a necesitar su uso.

	Perder el acceso a los servicios externos de los que haga uso la aplicación.	Minimización: Que no sea dependiente de ésta una funcionalidad principal.
		Plan de Contingencia: Tener en cuenta alternativas similares.
Requerimientos	Requerimientos de última hora.	Prevención: Tener en cuenta tiempo de reacción en la planificación.
		Minimización: Incrementar el tiempo de trabajo y la productividad.
	Cambios en requerimientos.	Prevención: Cambios en la aplicación o en la documentación.
		Minimización: Que no sean requerimientos básicos los afectados.
Otros	Que se presente en el mercado una idea o proyecto alternativo para solucionar el mismo problema.	Minimización: Estudiar el nuevo proyecto y ver qué se puede mejorar o seguir con el desarrollo y presentar y defender lo obtenido sin tener en cuenta la nueva idea.

Tabla 2. Análisis de estrategias con respecto a los posibles riesgos

3. Planificación

La planificación del proyecto es uno de los apartados más importantes. Ayuda a gestionar mejor el tiempo que vas a disponer para el desarrollo del proyecto y, como he previsto y analizado en el Análisis de riesgos, una buena planificación puede darte tiempo para reaccionar ante imprevistos.

Aunque generalmente, durante el desarrollo de un proyecto, las fechas de la planificación e incluso la planificación misma van cambiando, es bueno tener una base de la que partir. Y muchas veces los cambios en ésta no implica que vayan a afectar negativamente o sean debido a contratiempos, pueden ser debidos a la mayor experiencia que se va cosechando durante el transcurso del trabajo.

En la Tabla 3, se puede ver la planificación del tiempo que será empleado en desarrollar cada contenido y la fecha límite para tenerlo terminado.

Contenidos	Tiempo total	Fecha límite fin
Motivación, justificación y objetivo general Introducción Estudio de viabilidad Planificación Estado del arte	2 semanas	12 febrero
Objetivos Metodología Análisis y especificación Presupuesto, estimaciones, planificación	2 semanas	26 febrero
Diseño	2 semanas	12 marzo
Implementación	2 meses	7 mayo
Pruebas y validación Resultados Conclusiones y trabajo futuro Referencias, bibliografía y apéndices Agradecimientos, citas, índices	2 semanas	21 mayo

Tabla 3. Planificación temporal TFG

Esta planificación es debida a que, ya habiendo terminado el proyecto ABP del 4º curso de carrera, este año solo tendría que hacer el TFG y hacer las dos asignaturas de prácticas de empresa, equivalentes a 300 horas.

En principio, la planificación me permite acabar el TFG de cara a la convocatoria de junio, haciéndolo a la vez que curso las prácticas en empresa.

Si se diese el caso que necesitase más tiempo, como he previsto en algunos casos en el Análisis de riesgos, podría aumentar las horas de trabajo diarias, que según la planificación deberían ser aproximadamente poco más de 3 horas diarias para lograr el objetivo en las fechas establecidas. Incluso si necesitase de más tiempo todavía, podría plantearme la defensa en la convocatoria de julio.

4. Estado del arte

En este apartado se busca hacer una investigación minuciosa y profunda de los temas relacionados con la idea base del proyecto, así como conocer más información sobre el problema y las soluciones que existen en la actualidad.

En los siguientes subapartados se investigará sobre la geolocalización y su uso actual, diferentes soluciones tecnológicas a distintos problemas que usen la localización o sistemas similares en su tecnología, la forma legal del uso de la localización o ubicación en teléfonos móviles y las distintas plataformas que ofrecen mapas para usarlos de forma gratuita en tu app.

4.1. Localización en móviles

Para empezar, como la aplicación va a depender prácticamente en su totalidad de la localización, en principio de teléfonos móviles, voy a investigar sobre la localización: qué es y su uso en smartphones.

La geolocalización es una tecnología que utiliza datos obtenidos de la computadora o dispositivo móvil de un individuo para identificar o describir su ubicación física real.

Un sistema de geolocalización es una solución de la tecnología de la información que determina la ubicación de un objeto en un entorno físico o virtual. [5]

Aunque pueden existir diferentes fines o propósitos para utilizar o interactuar con la localización, como buscar servicios en un mapa o aportar información geográfica a un objeto, se pretende en el proyecto usar la localización con la finalidad de determinar una ubicación física de un objeto, como puede ser un smartphone o un localizador, que indique la posición de la persona que lo porta con relación a un sistema de coordenadas, en un mapa, para poder conocer información más específica a partir de la obtenida.

Además de obtener la localización a través de un teléfono móvil o un dispositivo GPS, también hay más opciones de recopilar esa información como puede ser a través de las direcciones IP durante la navegación web y el posicionamiento basado en WiFi.

4.1.1. Sistema A-GPS

Concretamente, el GPS de un smartphone funciona con un sistema denominado A-GPS (Assisted Global Positioning System), un sistema que combina el posicionamiento GPS de un móvil, en el cual se envía y se recibe información constante de una red de aproximadamente 30 satélites, con la recepción de datos que recogen las antenas de telefonía a las que el smartphone se conecta, con la finalidad de ahorrar en el consumo de batería y datos.

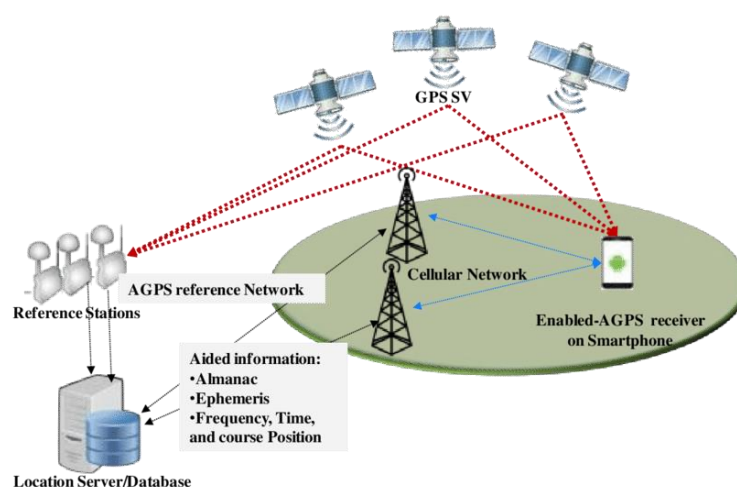


Figura 3. Esquema del funcionamiento del sistema A-GPS
(Fuente: https://www.researchgate.net/figure/Figure-2-16-AGPS-overview-system-representation_fig14_314229102)

4.1.2. Posicionamiento basado en WiFi

Éste se utiliza como alternativa a la localización GPS cuando ésta no es adecuada. Este método aprovecha los puntos de acceso inalámbricos en áreas urbanas utilizando la intensidad de la señal que recibe el smartphone.

El sistema de posicionamiento basado en redes inalámbricas WiFi, también llamadas redes Wireless, se basa en las mediciones que los nodos satélites de acceso a la red hacen de la potencia y de la relación SNR, relación señal/ruido, de las emisiones transmitidas por el nodo cliente inalámbricos que se conecta a la red. [6]

Es común que este sistema se pueda utilizar combinado con el sistema GPS.

4.1.3. Por direcciones IP

En primer lugar, se va a hablar de las direcciones IP y cómo se componen o se generan. Hay de dos tipos IPv4 o IPv6, según tengan cuatro bloques de números separados un punto entre ellos u 8 bloque de números y letras combinados separados por dos puntos entre ellos. Estas combinaciones no indican ningún tipo de localización como podría indicar un prefijo en un número de teléfono o un código postal, pero si aportan información sobre el proveedor que la genera o a grandes empresas.

Este tipo de geolocalización localiza la IP pública, que se muestra a la red de forma abierta cuando te conectas a internet, aunque cada vez que te conectes ésta cambiará ya que es dinámica.

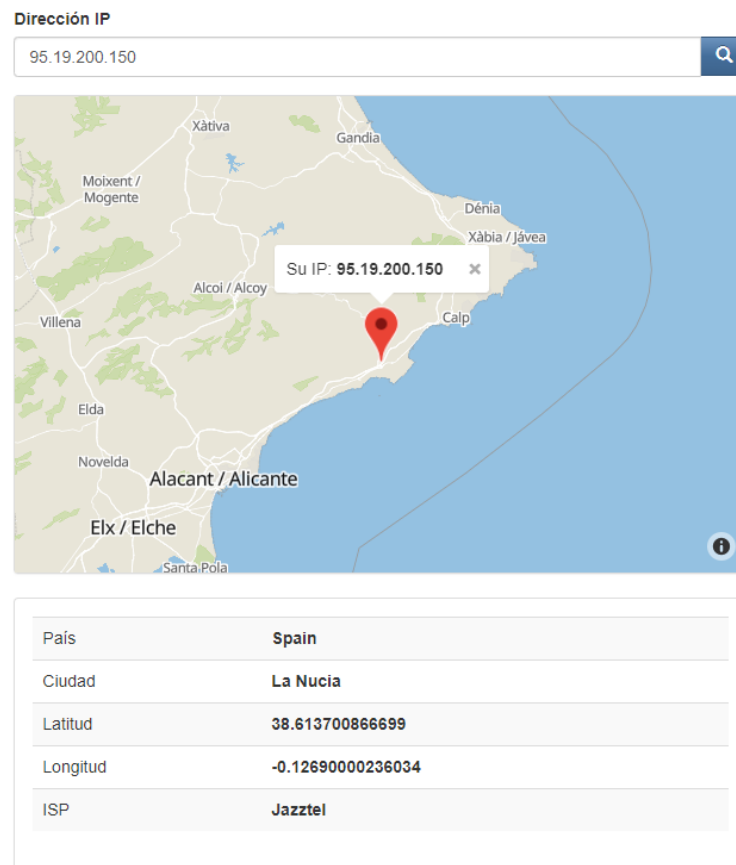


Figura 4. Ejemplo de geolocalización por IP online
(Fuente: <https://www.cual-es-mi-ip.net/geolocalizar-ip-mapa>)

Habitualmente, el resultado de esta geolocalización es la ubicación del proveedor de Internet, que es el que se encarga de asignarla a tu dispositivo, aunque la localización no sea precisa es un punto cercano al lugar exacto que se quiere localizar.

Aunque también, teniendo en cuenta los nodos de acceso telefónico, sí se puede conseguir una localización más exacta con una IP pública.

4.1.4. Geolocalización GSM

La geolocalización GSM o 2G (segunda generación) es un servicio ofrecido por las empresas operadoras de telefonía móvil que permite determinar, con cierta precisión, donde se encuentra físicamente un terminal móvil determinado.[7]

Tiene lugar gracias a las torres y antenas de telefonía. Su principal cometido es proporcionar cobertura a los dispositivos móviles. Además, estas antenas son capaces de rastrear un smartphone para geolocalizarlo, aunque de forma no del todo precisa, pueden marcar la zona donde se encuentra el dispositivo pero no su ubicación exacta. Su margen de error podría llegar hasta los 200 metros.

4.2. Ejemplos de utilización de la localización

4.2.1. En vehículos de transporte de mercancías

Generalmente enfocado a coches y camiones, se utilizan dispositivos o aplicaciones GPS que localizan el vehículo para recomendarle una ruta a seguir o bien para ver que ruta ha seguido o donde se encuentra.

En camiones, estos servicios suelen ser contratados por empresas de transporte de mercancías. Su objetivo es localizar sus vehículos en ruta y, además, poder consultar tiempos de trayecto y de paradas, previsión de duración del viaje, una posible vía de comunicación con el conductor del camión, la ruta que han realizado hasta el momento y la ruta recomendada a seguir según su destino.

Existen 3 diferentes tipos de localizadores en el mercado que se utilizan en el rastreo de vehículos, aunque también puedan ser utilizados enfocados a otros ámbitos. Estos tipos son:

- Localizadores GPS con conexión GPRS o 3G.
También pueden ser denominados GPRS (General Packet Radio Service), si usa tecnología 2G, o UMTS (Universal Mobile Telecommunications System), en el caso de usar 3G.

Estos tipos de localizadores usan la misma infraestructura de comunicación que las empresas de telefonía móvil, por lo que el dispositivo GPS disfrutará de buena cobertura. Necesitan para su funcionamiento de una tarjeta SIM y, en smartphones u otros, puedes hacer transmisiones de datos o envíos en forma de SMS para notificar la posición. Gracias al gran ancho de banda del que disponen, estos dispositivos son capaces de hacer grandes envíos de información.

En contra de todas estas características, estos tipos de localizador consumen mucha batería y pueden ser inhabilitados fácilmente.

- Localizadores GPS LPWAN.

Estos localizadores son muy similares a los de conexión GPRS o 3G mencionados anteriormente, aunque hacen uso de la tecnología LPWAN Low-Power Wide-Area Network en lugar de utilizar una tarjeta SIM. Esta tecnología está pensada y enfocada para IoT, “Internet of Things”, ya que se pretende que los dispositivos se comuniquen entre sí a través de la misma frecuencia.

Tienen una gran cobertura como los GPRS, aunque su consumo de batería es menor y estos dispositivos son inmunes a inhibidores de frecuencia. Pueden transmitir muy poco volumen de datos debido a su poco ancho de banda.

- Localizadores Bluetooth.

Estos localizadores, al no llevar GPS incorporado, son dependientes de un teléfono móvil que si que incorpore GPS para poder determinar una ubicación.

Tienen un coste y tamaño muy reducido comparado con el resto de los localizadores mencionados y solo pueden localizar en rangos de aproximadamente 10 metros, siendo estas localizaciones menos precisas si se encuentran en interiores.[8]

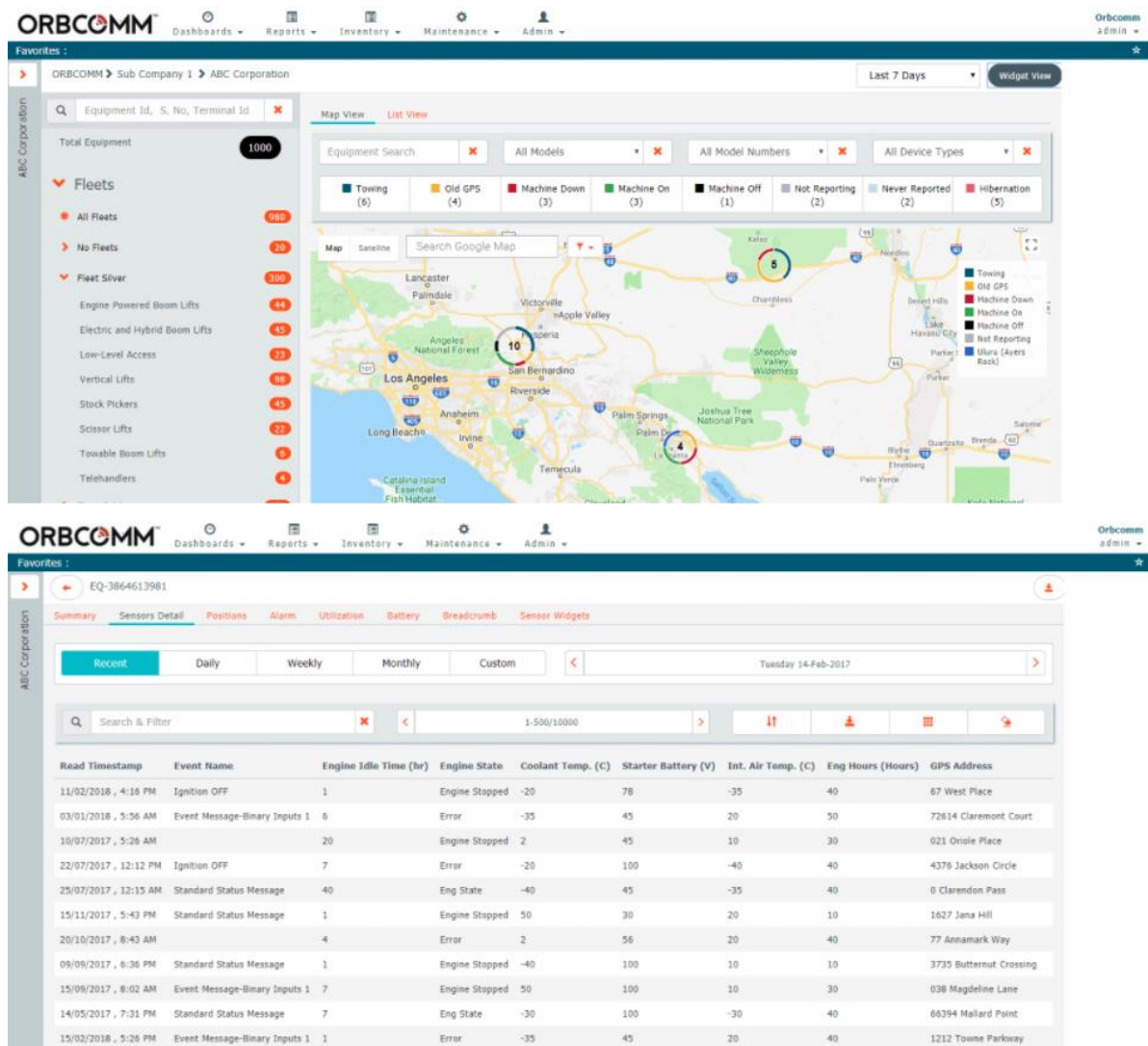


Figura 5. Interfaces de aplicación web para seguimiento, monitoreo y control de maquinaria pesada
(Fuente: <https://www.orbcomm.com/es/industries/heavy-equipment>)

4.2.2. Antirrobo

Cuando buscas en internet sistemas de localización o GPS antirrobo, los resultados siempre te dirigen a GPS antirrobo para vehículos tales como coches o motos.

Normalmente estos dispositivos antirrobo también incluyen otras funcionalidades enfocadas al rastreo del vehículo post robo, como puede ser la propia localización simultanea o un aviso cuando se sale de un perímetro establecido, o al impedimento del propio robo, como sería que esta evite que el motor pueda ponerse en marcha.

Hay diferentes tipos de GPS para estos fines, como se han mencionado en el punto anterior se pueden llegar a encontrar que funcionen con o sin tarjeta SIM.

4.2.3. En aves y otros animales

El uso de la localización u otros tipos de tecnologías en el campo de estudio de los recorridos y comportamiento de las aves migratorias es gran utilidad de cara a mejorar la información obtenida durante años con el método del marcaje de aves con anillas, o anillamiento, pudiendo contar con sistemas de seguimiento más precisos[9]. Estos ayudan a conocer detalles como cuánto tiempo permanecen las aves en sus áreas de cría e invernada, cuándo inician su migración, dónde las realizan, las velocidades y altitudes que alcanzan, cómo les pueden afectar distintos agentes meteorológicos que puedan suceder, cómo los relieves del terreno o mares o desiertos condicionan sus movimientos, qué puntos y hábitats utilizan para reponer energía. Todos estos aspectos permitirán ayudar en la conservación de estas.

Para lograr obtener esa información se trabaja actualmente con tres tipos de dispositivos: el emisor satélite, el data logger GPS y el geolocalizador.

El emisor satélite registra la localización geográfica y la emite en forma de señal captada por el sistema de satélites ARGOS y reenviada a un centro de recepción que procesa los datos. Los emisores satélites pueden llevar o no GPS incorporado, si lo llevan estos recogen coordenadas GPS que también son enviadas en la señal.

El data logger GPS, que tiene GPS incorporado como su nombre indica, recogen y acumulan la información, pero esta no es enviada. Esto provoca que la captura del animal sea necesaria para la descarga de la información que ha recabado el dispositivo. Actualmente, ya hay varios dispositivos que facilitan la descarga de datos sin necesidad de captura, solamente por cercanía.

El geolocalizador estima la posición geográfica a partir información que recibe como la hora de amanecer, la de anochecer como el número de horas diarias de luz con un sensor de luz incorporado, cuya función es tomar mediciones de esas variables cada poco tiempo. Con esta información se estima la latitud con la duración de día y noche y la longitud en base a la hora UTC del mediodía y de la medianoche.[10]

Otro uso de la localización para animales se enfoca a animales domésticos, como pueden ser los perros. Se trata de un dispositivo de seguimiento y rastreo de la mascota en tiempo real, similar al de los GPS para vehículos.

Este funciona con un chip receptor de emisiones satélites que permite, vía móvil o vía internet, conocer la localización de la mascota en caso de pérdida de esta. También pudiendo incluir

algunos de estos dispositivos la opción recibir alertas vía app o SMS en caso de que el perro saliese de una zona marcada restringida, como puede ser la zona de su rutina habitual de paseo.

De estos tipos de localizadores para mascotas pueden existir con chip GSM y tarjeta SIM de datos, que dependerán de la señal telefónica de un teléfono móvil y la necesidad de tener una app descargada, o sin tarjeta SIM, que en este caso tendrá una cobertura más limitada y se alimentará de una señal de radio convencional.

4.2.4. En pulseras para presos

Anteriormente, estas pulseras o tobilleras funcionaban con radiofrecuencia para así comprobar si el delincuente se encontraba en un lugar determinado donde había una unidad que recogía la señal. La unidad recogía cualquier tipo de información, como nivel de batería o corte de la correa, pero solo si ocurría cuando el delincuente se encontraba cerca de esta.

En la actualidad se utilizan brazaletes GPS con SIM integrada, que pueden rastrear al sujeto donde sea que se encuentre y comunicar las infracciones instantáneamente. Estos brazaletes también funcionan con una unidad doméstica.

4.3. Soluciones existentes

Actualmente, en el mercado existe una gran variedad de aplicaciones o servicios que ofrecen una solución y que serían alternativas al proyecto que se plantea en este trabajo. En este apartado se van a estudiar a varias de ellas para poder conocerlas mejor, ver qué ofrecen y así poder mejorar y coger ideas de forma previa a trabajar en el diseño y las especificaciones.

En la selección de opciones a estudiar se van a estudiar de diversas características como pueden ser aplicaciones o servicios que exijan la necesidad de comprar un dispositivo GPS, otras que solo necesitan smartphones, que estén dirigidas solo a un tipo de usuario específico u otras más abiertas pensadas para poder localizar a cualquier persona sin características concretas.

Para lograr encontrar estas aplicaciones o empresas que ofrecen estos servicios se han hecho búsquedas a través de la tienda de aplicaciones de Google, Google Play, y del propio buscador web de Google en el navegador.

Estas búsquedas han sido realizadas con los términos “aplicación localizador”, “localizadores”, “localizadores para mayores”, “localizadores para niños” o “GPS para personas mayores”. La elección de estas aplicaciones ha sido acorde a un límite de valoración que he puesto personalmente en 4 sobre 5 y permitiendo alguna que tenga menos valoración debido a que son aplicaciones oficiales.

Con todo lo explicado anteriormente, en los siguientes subapartados se explicarán las aplicaciones seleccionadas.

4.3.1. LoPe (Localizador de personas)

LoPe (localizador de personas) es un servicio ofrecido por la Teleasistencia de Cruz Roja. Este servicio busca fomentar la independencia e incrementar la seguridad de la persona que lo porta, así como tranquilidad a familiares o personas cercanas que estén en constante alerta por ellos. Gracias a esta aplicación se puede comprobar en todo momento la ubicación y el trayecto que ha recorrido el usuario del servicio.[11]

Este servicio, no disponible en tiendas de aplicaciones como Google Play, va dirigido a personas que salen solas a caminar, a deportistas, personas que puedan tener necesidades especiales y quieran ser autónomas e independientes, personas que pueden desorientarse o perderse o encargados de personas con deterioro cognitivo.

Este servicio funciona a través de una aplicación que se instala en un smartwatch, el cual envía la posición cada 10 minutos con su localización geográfica y permitiendo configurar alarmas de entrada y salida en áreas marcadas, así como alarmas de velocidad.

Cuando surge la necesidad de localizar a la persona que porta el localizador, el sistema se activa, así como en caso de que el terminal envíe un aviso automático, el cual facilitaría una respuesta y una intervención ante cualquier necesidad o eventualidad.

También, en caso de emergencia, Cruz Roja avisa a familiares o personas de contacto y a servicios de emergencias, como médicos, policía o bomberos, para una pronta atención, facilitando a quien procede sus datos personales y de salud.

Este servicio no es gratuito, se contrata en Cruz Roja y tiene un coste, según la web, de menos de 1€ al día.

4.3.2. Weenect

Weenect ofrece una gran gama de localizadores para toda la familia, como mayores y niños, e incluso para mascotas, como perros y gatos.[12] Este servicio tiene una valoración de 4,4/5 en Google Play y de 4,1/5 en Trustpilot.[13]

Con Weenect se pueden consultar posiciones sin límite, permite tener más de un usuario y más de un localizador en una misma cuenta e interfaz y su rango de distancia es seguimiento es ilimitado (funciona en más de 100 países sin coste adicional).

También incluye un botón de SOS, permite la llamada al portador del localizador a través de un altavoz en este, notifica salidas de zonas marcadas como seguras o habituales y también contiene un historial de ubicaciones y de seguimiento de actividad.

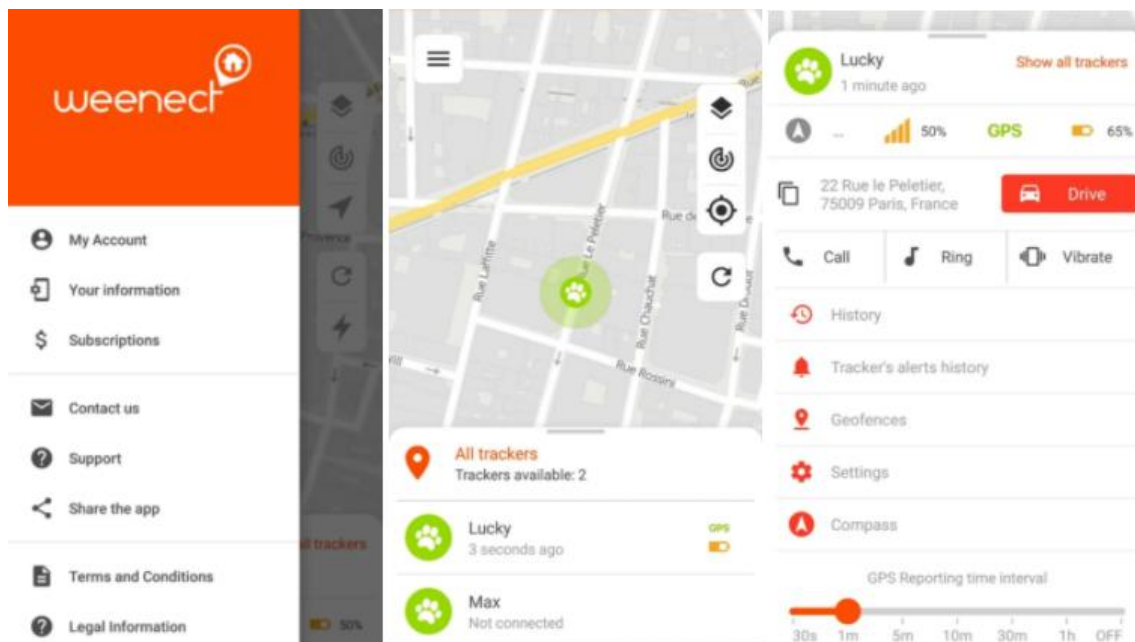


Figura 6. Interfaces de Weenect

(Fuente: <https://play.google.com/store/apps/details?id=com.maptitebalise.MPB&gl=ES>)

Todas las alertas de la aplicación se pueden configurar como notificación en el smartphone, como email o como SMS.

Este servicio tiene el coste que implica la compra del localizador específico, ya sea para ancianos, niños, perros o gatos todos con coste de 49,90€ aunque con características distintas, añadido a una suscripción fija en todas las opciones de 3,75€ mensuales.

4.3.3. Find My Kids

Find My Kids es una aplicación enfocada a localizar a tus hijos instalando una app especial en su móvil o conectando un reloj GPS. Este servicio tiene una valoración de 4,6/5 en Google Play.[14]

Para poder empezar a hacer uso de la aplicación se necesita descargar también la app “Pingo: chat with parents” disponible en tiendas de aplicaciones[15] o opcionalmente usar un reloj-GPS.

En cuanto a la aplicación Pingo, tiene una valoración de 2,9/5 en Google Play con un gran número de opiniones divididas. Generalmente los padres opinan que funciona correctamente y los hijos comentan que la herramienta es usada por padres como espionaje y ellos anulan totalmente su privacidad.

Esto se debe a que, entre las diversas funcionalidades, la aplicación permite escuchar en cualquier momento lo que está ocurriendo en el móvil de tu hijo, enfocado según la empresa desarrolladora en su web para “escuchar lo que ocurre alrededor de tus hijos cuando no respondan a tus llamadas o estén en mala compañía”. Esta función no estaría disponible para iPhone.[16]

Otras funcionalidades que ofrece la app son usar el GPS localizador para consultar la ubicación e historial de movimientos, poder enviar un sonido fuerte al otro móvil, controlar qué aplicaciones ha usado en el smartphone, poder consultar el nivel de batería del teléfono, recibir mensajes SOS del niño o habilitar un chat privado para comunicarte a través de la app.

El servicio que ofrece esta aplicación es gratuito, se pueden comprar pulseras reloj con localizador, pero es un elemento complementario y opcional por lo que no sería necesario para el funcionamiento.

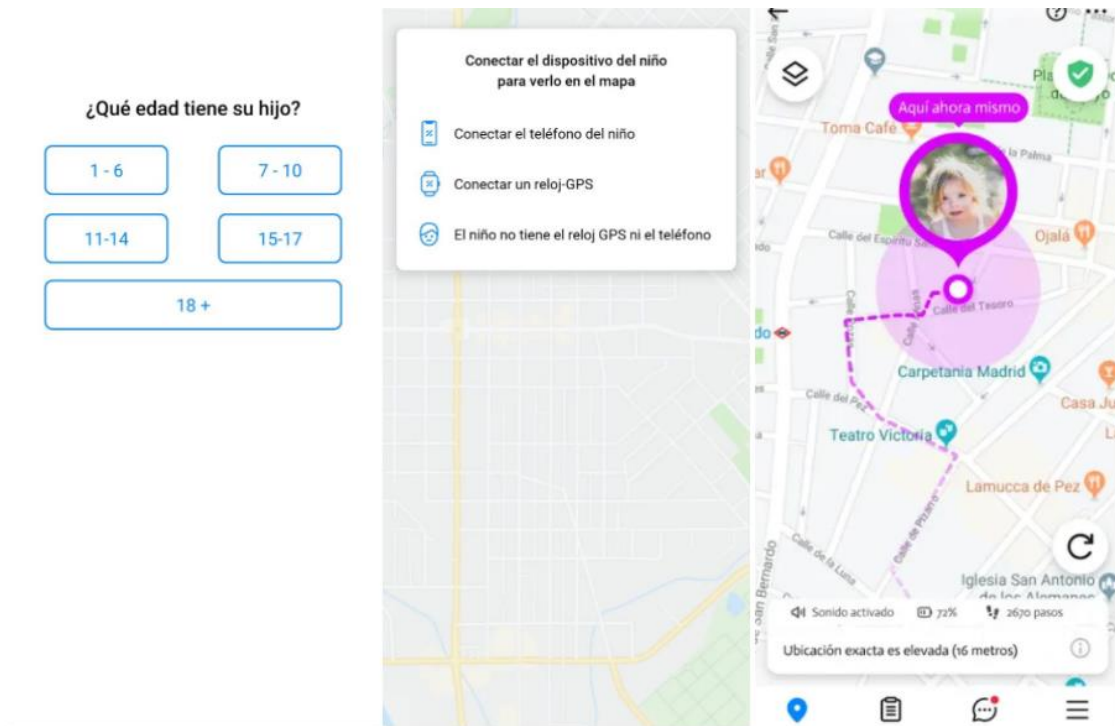


Figura 7. Interfaces de Find My Kids
(Fuentes propia y <https://findmykids.org/es>)

4.3.4. Life360

Life360 es una aplicación que funciona como localizador de familia. La aplicación tiene una valoración de 4,4/5 en Google Play.[17]

Esta aplicación, destinada y dirigida al ámbito familiar completo, permite con sus funcionalidades ver la ubicación de los miembros de tu familia en un mapa privado accesible por invitación, elegir cómo compartir tu ubicación con cada grupo, chats individuales y grupales privados, recibir alertas o notificaciones cuando un familiar llegué a un destino o rastrear un teléfono perdido o robado.

El servicio que ofrece la aplicación es gratuito, aunque tiene un plan premium de 4,17\$ mensuales, aproximadamente 3,46€, con el que puedes tener alertas ilimitadas de lugares, informes de conducción familiares y personales, historial de ubicaciones de los anteriores 30 días y atención al cliente prioritaria.

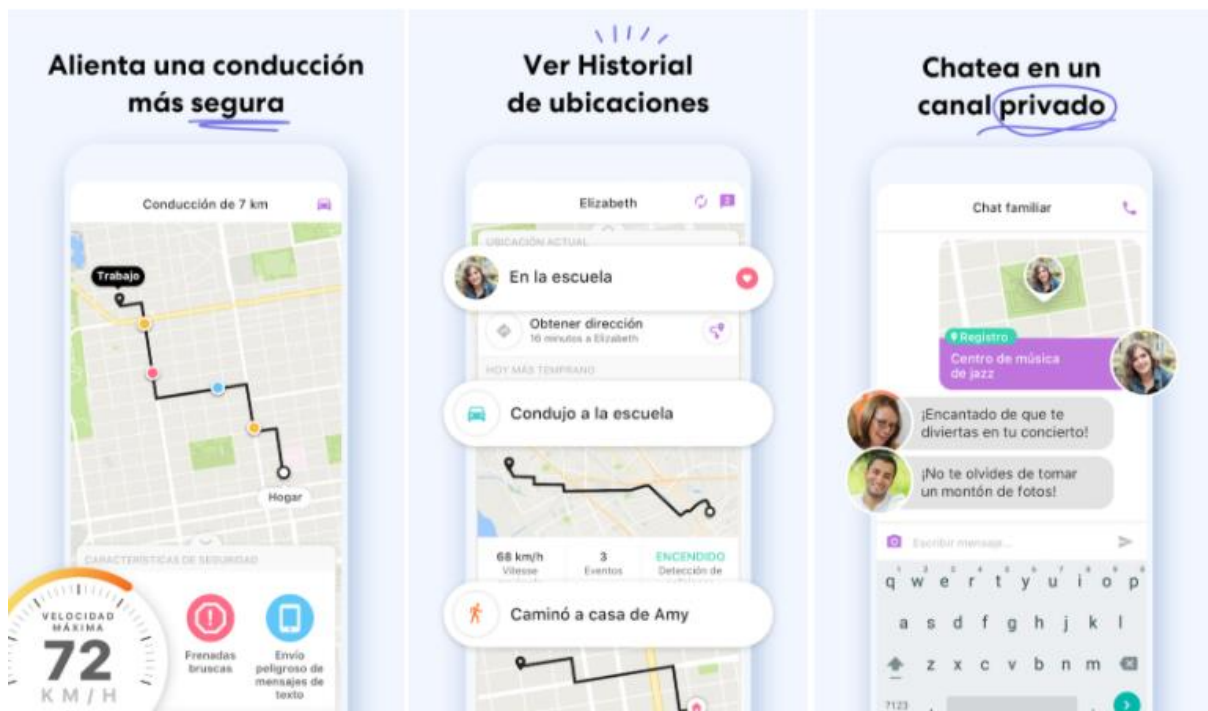


Figura 8. Interfaces de Life360

(Fuente: <https://market.android.com/details?id=com.life360.android.safetymapd>)

4.3.5. GeoZilla

La aplicación de GeoZilla es un rastreador de teléfonos GPS en tiempo real con seguimiento de ubicación multiplataforma.[18] Su valoración en las tiendas de aplicaciones es 3,8/5 en Google Play y 4,5/5 en Apple Store.[19]

La app de GeoZilla permite que se notifique o alerte cuando un familiar entre o sale de casa, el uso de algoritmos “amigables” con la batería para no exceder el gasto de esta, un chat privado ya sea individual o grupal incluso con capacidad para compartir contenido multimedia en él, poder conseguir el historial de ubicaciones tuyo o de tus familiares.

De forma inicial, el servicio que ofrece GeoZilla es gratuito, aunque también contiene un plan Premium contratable de 47,99\$ al mes, aproximadamente 40,08€, 79,99\$ anuales, unos 66,80€, o 5,99\$ semanales, que equivaldrían a 5€, aunque la misma empresa indica que estos precios pueden variar según la ubicación en la que uses su aplicación. Este plan premium de pago ofrece la posibilidad de un número ilimitados de alertas en ubicaciones, el historial de ubicaciones completo disponible para 2 semanas y un horario familiar ajustable.



Figura 9. Interfaces de GeoZilla
(Fuente: <https://play.google.com/store/apps/details?id=com.geozilla.family>)

4.3.6. AlertCops

AlertCops es la app de la Policía y la Guardia Civil, el canal discreto y eficaz para solicitar ayuda de las Fuerzas y Cuerpos de Seguridad del Estado, un proyecto del Ministerio del Interior. Su valoración en la tienda de aplicaciones es 4,0/5 en Google Play.[20]

Es una aplicación más amplia, no únicamente la funcionalidad de la localización de personas, aunque como también la incluye y es una aplicación oficial vale la pena estudiarla.

Este servicio incluye la posibilidad de chat con tu centro policial más cercano, recibir avisos si hay una emergencia de seguridad en la zona en la que te encuentras y un botón de SOS para alertar de forma rápida a las autoridades. Además, la aplicación incorpora un traductor automático para poder comunicarte en más de 100 idiomas.

La funcionalidad que interesa investigar se denomina Guardián, una funcionalidad que permite siempre que quieras poder compartir tu posición, que se enviaría periódicamente, con los tuyos o con los servicios de rescate. Esto ayuda a que, en caso de emergencia, tu localización sea más rápida y precisa.[21]

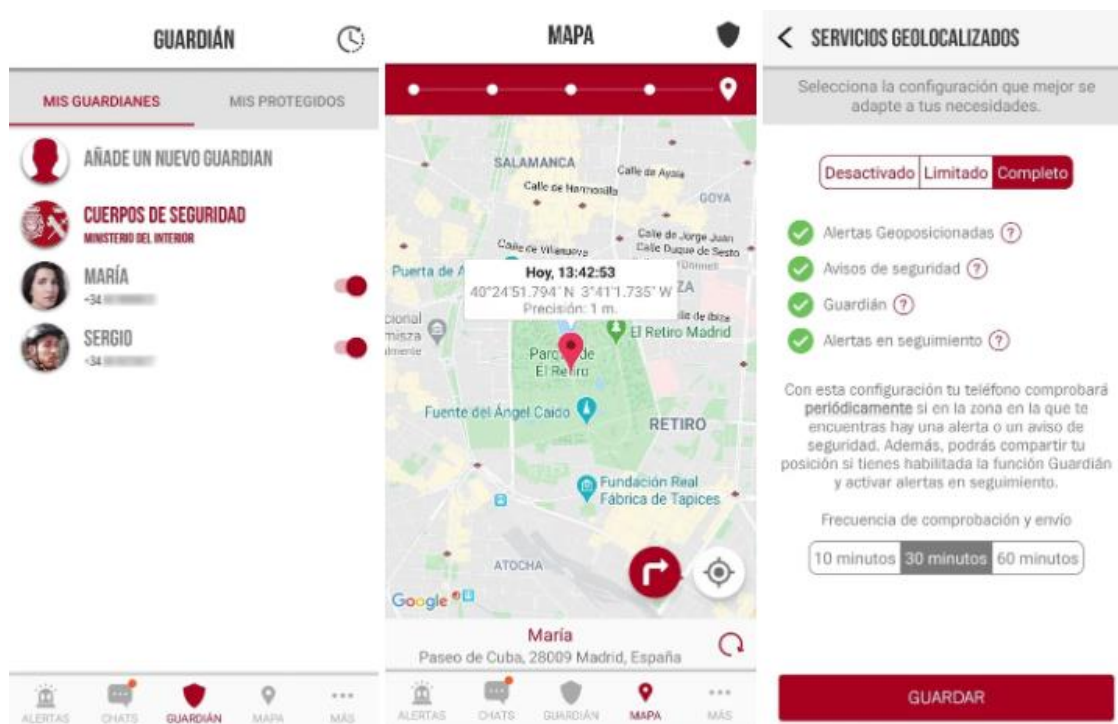


Figura 10. Interfaces de AlertCops y Guardián
(Fuente: <https://play.google.com/store/apps/details?id=com.alertcops4.app&hl=es>)

Al ser un servicio oficial y estatal, esta aplicación es de uso gratuito.

4.3.7. Conclusiones

Tras conocer más exhaustivamente otras soluciones existentes al problema, ahora se puede hablar y comparar lo que aportan unos u otros que puede ser más o menos interesante de cara a tenerlo en cuenta en diseño y definición de requisitos, objetivos y funcionalidades y posterior diseño de la misma aplicación.

Estás aplicaciones, salvo LoPe, tienen en común una interfaz principal la cual muestra un mapa con la ubicación del elemento a localizar. Algunas de estas aplicaciones implican la necesidad de compra de un localizador, otras localizan teléfonos móviles y otras permiten la adquisición de pulseras o relojes inteligentes, en ocasiones como ampliación de la localización de un smartphone.

En base a la idea inicial, se buscaba que una de las características principales fuese el aviso o notificación cuando la persona localizada sale de un área o zona delimitada. Esta funcionalidad también está incluida en la mayoría de las aplicaciones y soluciones estudiadas, hecho que reafirma la decisión de que sea la funcionalidad principal y base del trabajo.

Otras funcionalidades de las aplicaciones analizadas que pueden ser tenidas en cuenta son la inclusión de un botón de emergencia o SOS, la posibilidad de consultar un historial de ubicaciones o incluir un sistema de aviso a servicios de emergencia como podrían ser policía, bomberos o ambulancias.

5. Objetivos

Tras una investigación para conocer más el tema sobre el que tratará el TFG, en este apartado se van a identificar y exponer los objetivos que se buscan obtener en el transcurso del proyecto y el desarrollo de la aplicación.

Estos objetivos y subobjetivos ayudarán a poder marcar metas a conseguir más realistas y alcanzables, al igual que podrán ser de ayuda para terminar de definir las funcionalidades que se vayan a implementar y con las que contará la aplicación.

Para hablar de estos objetivos, se va a hacer con el método creado por George T. Doran como objetivos SMART, acrónimo de Specific, Measurable, Achievable, Realistic y Timely, que indica las cualidades que debe tener un objetivo para ser claro y preciso. Se buscan objetivos específicos, medibles, alcanzables, realistas y con un límite de tiempo razonable y realista para llevarlos a cabo.[22]

El objetivo principal de la aplicación es comprobar cuando una persona sale de una zona y poderlo notificar y mostrar en una interfaz sencilla con un mapa. A partir de este objetivo principal, se pueden definir otros subobjetivos para la aplicación:

- Que el usuario de la aplicación pueda elegir la zona de seguridad del usuario localizado delimitándola en un mapa.
- Que el usuario pueda desconectar y conectar libremente la notificación o alarma que indica las salidas de zona.
- Poder previsualizar en un mapa la última posición conocida guardada en base de datos si ha salido de la zona marcada.
- Que más de un usuario pueda seguir al mismo usuario localizado con opciones y preferencias distintas.
- Que la aplicación sea sencilla y tenga el mínimo número de interfaces posible para un uso correcto y adecuado sin necesidad de hacer un número excesivo de clicks.

También es interesante definir con este método unos objetivos más bien relacionados con el TFG, no únicamente con la aplicación, como por ejemplo objetivos de aprendizaje o experiencia, que puede ser útil como metas personales a alcanzar durante el desarrollo de este:

- Lograr empezar y acabar una aplicación funcional completa en la que he trabajado únicamente yo.
- Poder crear y administrar una Base de Datos propia.
- Que el resultado final de la aplicación sea completamente tangible y de buena calidad, por lo que también podría ser incluida como parte de mi portfolio personal.

6. Metodología

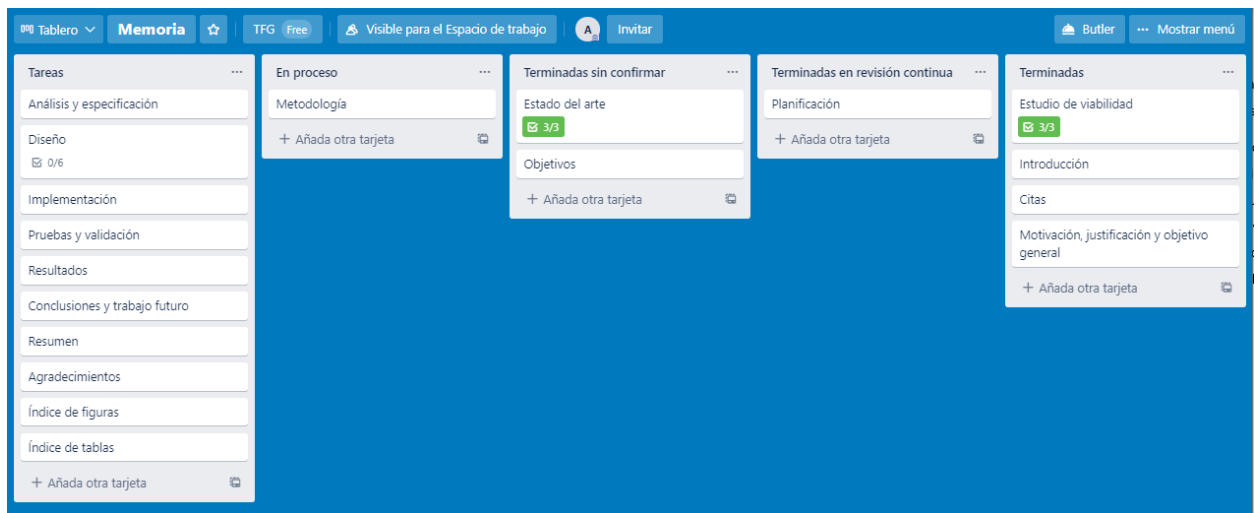
En este apartado se va a describir las metodologías a utilizar durante el desarrollo del proyecto TFG por completo, ya sea utilizadas hasta el momento o las que vayan a ser utilizadas en apartados posteriores o en fases de implementación, por qué han sido elegidas y que aportan al desarrollo de este. Para el proyecto han sido elegidas dos metodologías Agile para combinarlas a la hora del desarrollo e implementación del TFG.

Para el desarrollo escrito de la memoria del proyecto y sus apartados ha sido elegida la metodología Kanban.

Esta metodología ayuda a hacer más visible el flujo de trabajo, qué tareas se tienen que hacer en un trabajo y así gestionar correctamente como desarrollarlas, escribiendo estas tareas en un tablero estando organizadas en este según su estado.[23]

En el caso de este proyecto, el tablero Kanban ha sido dividido en las columnas “Tareas”, para las tareas que todavía no han sido iniciadas; “En proceso”, para las tareas iniciadas sin acabar; “Terminadas sin confirmar”, para aquellas acabadas pero que todavía no han pasado una revisión del tutor; “Terminadas en revisión continua”, para aquellas tareas acabadas pero que pueden sufrir cambios a lo largo de todo el desarrollo de la memoria y, por último, “Terminadas”, para las tareas ya finalizadas por completo.

Para poner en marcha el tablero Kanban del desarrollo de la memoria se ha utilizado la herramienta web gratuita “Trello”, en la que se puede personalizar libremente las tareas y columnas del tablero como se ve en la Figura 11, que muestra el estado actual del Kanban del TFG.



*Figura 11. Captura del tablero Kanban del proyecto en Trello
(Fuente propia)*

Para la implementación de la aplicación se ha optado por seguir una metodología Scrum.

Esta metodología se basa en la división del tiempo de desarrollo en fases, llamadas sprints o iteraciones, para las cuales se ponen objetivos a alcanzar y hacer revisiones periódicas para ver cómo se está trabajando y cómo de encaminados van los objetivos. Este tipo de metodología implica que hay una menor probabilidad de imprevistos durante el desarrollo.[24]

Aunque Scrum está enfocado al trabajo en equipo, se utilizará esta metodología sin tener en cuenta aquello relacionado con roles de equipo y reuniones, puesto que el trabajo es individual.

Además, otra herramienta utilizada durante el desarrollo del proyecto es la herramienta “Toggl”, que permite hacer un control del tiempo invertido en el proyecto y en cada una de las tareas. Esta herramienta también permitiría, por ejemplo, estar ligada al tablero Kanban para sumar los tiempos de trabajo realizados a las tarjetas o tareas de este a realizar. Esta herramienta también permite comprobar si los tiempos estimados al principio del desarrollo han sido acertados.

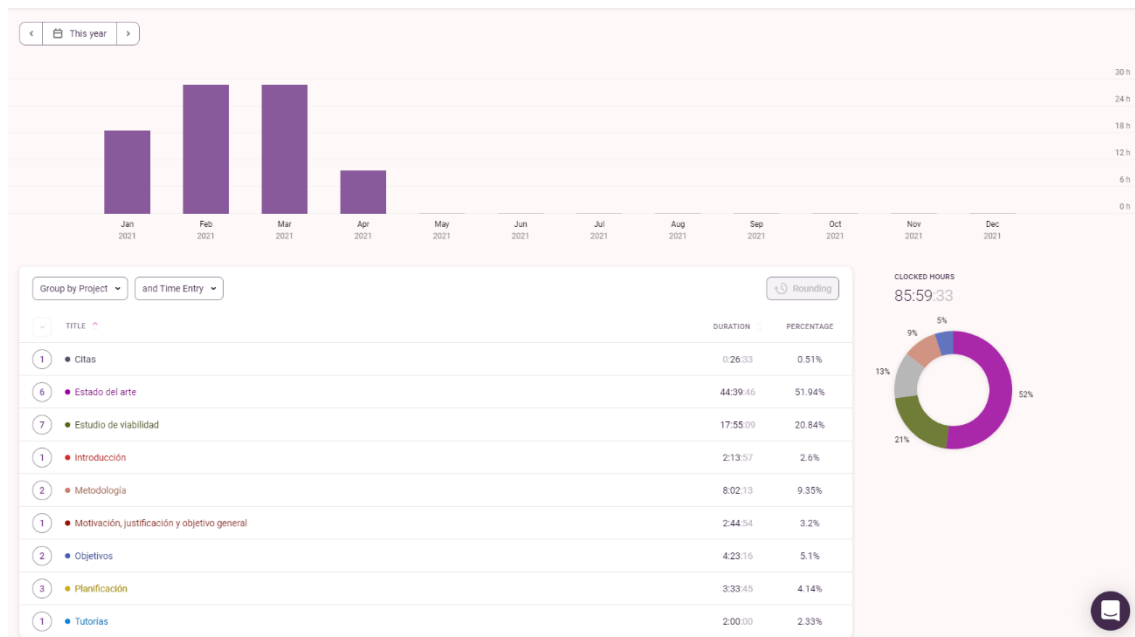


Figura 12. Captura del estado actual de la herramienta Toggl
(Fuente propia)

7. Análisis y especificación

Esta sección está enfocada al análisis de las necesidades que debe tener la aplicación para solucionar el problema al que está orientada, así como que requisitos y funcionalidades tendrá y a quien puede ir dirigida.

La información que se presentará en este apartado, definiciones de tipos de usuarios que tendrá la aplicación y especificación de requisitos de ésta, servirá como base para guiar las decisiones posteriores para la aplicación. El desarrollo del contenido del apartado y la especificación de requisitos se ha hecho siguiendo las recomendaciones y ayuda de la herramienta de análisis y especificación de estándar de IEEE 830.[25]

7.1. Tipos de usuarios

Para empezar, se describen los tipos de usuarios que existirán en la app, así como hasta donde llegarán sus capacidades de uso.

Tipo de usuario	Descripción
Usuario Guardián	Usuario general y básico, será el usuario principal y el que más uso dará a la aplicación. Podrá seguir a un usuario Protegido y configurar las zonas seguras y otros aspectos de la información de seguimiento que reciba.
Usuario Protegido	Este usuario no utilizará activamente la aplicación, su funcionalidad es enviar su ubicación a los usuarios Guardianes que le hagan el seguimiento. Podrá generar el código que permita que algún Guardián le siga.
Administrador	Administrador de la aplicación. Gestiona y se encarga de todo lo relacionado con los usuarios y del correcto funcionamiento de la aplicación, así como solucionar los errores que puedan sucederse.

Tabla 4. Tipos de usuario en la aplicación

7.2. Requisitos

En este apartado se van a detallar y definir todos los requerimientos para la aplicación del proyecto. Estos estarán divididos entre Requisitos funcionales y Requisitos no funcionales, a los cuales se les asignará un identificador, al igual que se indicará para que usuarios va dirigido cada uno.

7.2.1. Requisitos funcionales

Estos requisitos son aquellos que se reflejarán en la aplicación en forma de funcionalidades. Sus identificadores estarán formados por las iniciales RF, referente a su propio nombre.

Todos los requisitos funcionales serán de implementación obligatoria en el proyecto.

Identificador	RF1
Tipo de usuario	Guardián, Protegido
Descripción	Habilitar mecanismos mediante los cuales un Guardián pueda emparejarse y desemparejarse libremente con un Protegido y que únicamente sea posible conocer su posición mediante este emparejamiento.

Identificador	RF2
Tipo de usuario	Guardián
Descripción	Capacidad de elegir el rango o zona de seguridad del Protegido para delimitar la zona de uso de la alarma y posibilidad de edición de este rango libremente.

Identificador	RF3
Tipo de usuario	Guardián
Descripción	Capacidad para ver el mapa y el rango en el cual se encuentra el Protegido.

Identificador	RF4
---------------	-----

Tipo de usuario	Guardián
Descripción	Posibilidad de recibir notificaciones o alarmas cuando Protegido salga del rango seleccionado y de activar, desactivarla o configurarla libremente.

Identificador	RF5
Tipo de usuario	Guardián
Descripción	Posibilidad de cancelar la alarma tras recibir la notificación de salida de zona de un Protegido.

Identificador	RF6
Tipo de usuario	Guardián
Descripción	Posibilidad de consultar la última posición exacta guardada en base de datos de Protegido cuando esté se encuentre fuera de la zona de seguridad y haya saltado la alarma.

Identificador	RF7
Tipo de usuario	Administrador
Descripción	Gestión de los usuarios de la plataforma.

Tabla 5. Requisitos funcionales

7.2.2. Requisitos no funcionales

Los requisitos no funcionales son requisitos que debe tener en cuenta el proyecto para su planteamiento, diseño y funcionamiento, aunque no sean funcionalidades concretas de la aplicación. Sus identificadores estarán formados por las iniciales RNF, referente a su propio nombre.

Identificador	RNF1
Tipo	Distribución
Descripción	La aplicación estará disponible dentro de la tienda de aplicaciones Google Play y será de descarga libre y gratuita.

Identificador	RNF2
---------------	------

Tipo	Disponibilidad
Descripción	La aplicación deberá estar activa y con un correcto funcionamiento las 24h al día todos los días del año con disponibilidad del 99,9%.

Identificador	RNF3
Tipo	Usabilidad y accesibilidad
Descripción	<p>El diseño de las interfaces de la aplicación deberá ser adaptable para todos los tipos de pantallas de smartphones.</p> <p>El usuario deberá ser capaz de utilizar la aplicación sea cual sea su edad o conocimiento previo de esta, por lo que el diseño de interfaces también deberá ser intuitivo.</p> <p>Para un mejor uso de la aplicación, esta será diseñada para poder ser usada con el menor número de clicks posibles.</p>

Identificador	RNF4
Tipo	Legalidad
Descripción	El material utilizado para el desarrollo de la aplicación se encuentra con licencia válida para su uso.

Identificador	RNF5
Tipo	Eficiencia
Descripción	La aplicación tendrá el menor tiempo de respuesta posible, debido a la importancia de su correcto funcionamiento y poder recibir correctamente y a tiempo las alarmas correspondientes.

Identificador	RNF6
Tipo	Privacidad
Descripción	Los usuarios no podrán obtener más información de otros usuarios salvo su código de seguimiento, notificaciones de salidas y última posición

	conocida en caso de salida y perteneciendo esta siempre a un usuario Protegido con el cual está el enlace de seguimiento activo.
--	--

Identificador	RNF7
Tipo	Mantenimiento del código y futuras mejoras
Descripción	<p>El código será desarrollado para que en su forma final y definitiva pueda ser mejorado o modificado con facilidad, al igual que podrá estar disponible y listo para que sea posible añadir mejoras y nuevas funcionalidades en el futuro.</p> <p>Para un uso más seguro del código, se hará uso de un software de control de versiones.</p>

Tabla 6. Requisitos no funcionales

8. Diseño

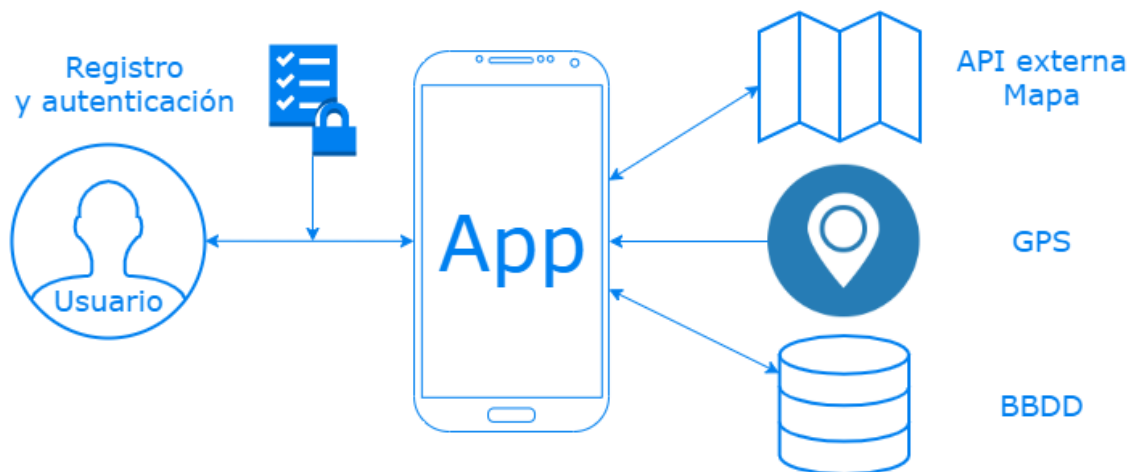
En este apartado, uno de los más importantes y con más peso en el proyecto, se va a desarrollar y explicar cómo se va a diseñar en distintos aspectos la aplicación a la que hace referencia el documento del TFG y cómo estos diseños buscan solucionar y cumplir con los requisitos mencionados en el apartado anterior.

Las decisiones y diseños mostrados en este punto serán una guía a seguir en la implementación final de la aplicación.

8.1. Diseño arquitectura conceptual

El diseño de la arquitectura conceptual del proyecto busca plasmar en una imagen en que elementos se dividirá el proyecto a grandes rasgos, sin concretar específicamente la tecnología a utilizar en cada uno.

Este esquema servirá como base al funcionamiento de la aplicación y las tomas de decisión posteriores.



*Figura 13. Diseño Arquitectura Conceptual
(Fuente propia)*

Todos los tipos de usuario descritos en la Especificación, en el punto anterior, tendrán acceso a la aplicación, la cual solo estará disponible para smartphones.

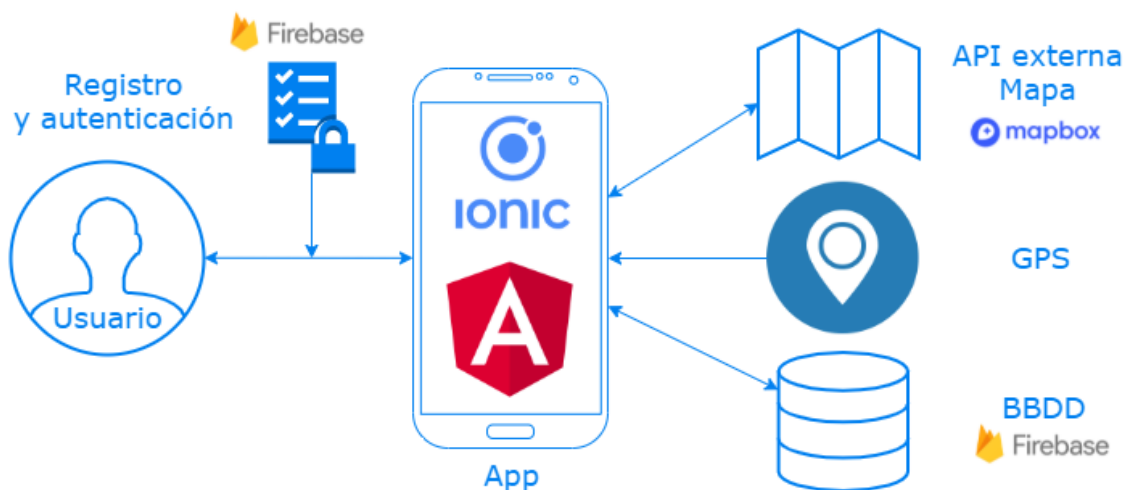
La aplicación hará uso de distintos servicios para ofrecer un funcionamiento completo al usuario:

- Facilitará un mapa en sus interfaces que será servido por una API externa, con la cual se podrán cumplir los requisitos RF2 y RF3 referentes a la selección de rango en el mapa y la visualización de este.
- Consultará la información necesaria sobre la ubicación del usuario Protegido a través del GPS del propio smartphone, pudiendo así notificar al Guardián cuando Protegido sale de la zona delimitada y cumplir los requisitos RF1 y RF4.
- Guardará toda la información estrictamente necesaria en una base de datos para poder servirla o consultarla por usuarios o Administradores cuando sea necesario, cumpliendo así los requisitos RF1 y RF6 en los cuales se generan mecanismos de seguimiento y se consulta la última posición guardada en la propia base de datos.

8.2. Diseño de la arquitectura tecnológica

Para este apartado, tomando como base el diseño de la arquitectura conceptual de la aplicación, se hará la propuesta de las tecnologías a utilizar para implementar cada una de las partes o bloques que compondrán la aplicación.

Teniendo en cuenta que la aplicación será exclusivamente para smartphones, el esquema de su arquitectura tecnológica será el siguiente:



*Figura 14. Esquema Arquitectura Tecnológica
(Fuente propia)*

Para implementar la aplicación se utilizará el framework¹ Ionic basado en Angular. El registro y la autenticación de usuarios en la aplicación estará desarrollado con el SDK² de Firebase, servicio ofrecido por el propio Google para desarrollo de aplicaciones.

Firebase también ofrece servicio de base de datos NoSQL y, a pesar de poder elegir entre distintos tipos, la opción más interesante para este proyecto que ofrece es Cloud Firestore, creada para el desarrollo de aplicaciones globales, con posibilidad de libre estructuración y consulta de esta, se puede acceder a ella y usarla sin necesidad de servidores y sincroniza los datos incluso sin conexión.

La lógica de la aplicación también será servida por el propio Firebase, desde el cual se accederá a su base de datos y al registro y autenticación de usuarios.

Por último, para servir los mapas de la aplicación se utilizará la única API externa de la arquitectura del proyecto, Mapbox.

8.3. Diseño de la persistencia

8.3.1. Almacenamiento de datos

En este apartado se mostrará las decisiones tomadas para el diseño de la base de datos, así como los campos o información a guardar en esta.

Como se menciona en el punto anterior, se hará uso de la base de datos Cloud Firestore del servicio Firebase de Google, que es una versión más reciente, ampliada y mejorada de la también ofrecida por Firebase, Realtime Database.

Inciendo más sobre Cloud Firestore, es una base de datos NoSQL. Estas bases de datos se basan en estructuras para poder almacenar información que no siguen el modelo entidad - relación ni hacen uso de tablas, más bien hacen uso de formatos como clave-valor o mapeos de grafos o columnas.

Como ventajas sobre SQL, las bases de datos NoSQL pueden ejecutarse sin apenas computación que facilita ejecutarlas en máquinas con menos recursos, gracias a su estructura distribuida tienen la capacidad de trabajar con una gran cantidad de información almacenada, es escalable

¹ Esquema para el desarrollo y/o implementación de una aplicación.

² Kit de desarrollo software (en inglés, software development kit).

horizontalmente y no genera los denominados “cuellos de botella” al recibir varias peticiones ya que para ejecutarlas NoSQL no necesita transcribirlas.

Las bases de datos NoSQL están más enfocadas a la lectura de datos que al almacenamiento o edición. Estas, al no ser relacionales, al actualizar datos previamente ya guardados requerirán cambiar uno a uno estos campos en el número de lugares en los que se encuentren. Debido a la rápida lectura de NoSQL, la aplicación será más rápida y el tiempo de respuesta menor cuando sirva estos datos.[26] [RNF5]

Volviendo al servicio de Cloud Firestore, es una base de datos flexible y escalable enfocada al desarrollo en servidores, dispositivos móviles y web que mantiene la sincronización de datos a tiempo real entre aplicaciones y ofrece servicio sin conexión. [RNF2]

Cloud Firestore permite la organización de estructuras de datos flexibles y jerárquicas y almacena la información en documentos, a la vez ordenados en colecciones. Estos documentos pueden tener a su vez asociadas subcolecciones, colecciones en niveles inferiores al de la raíz que nacen y están asociadas a un documento concreto.

También permite hacer consultas para obtener resultados de documentos concretos o de varios documentos de una misma colección y el rendimiento de estas consultas dependerá directamente del tamaño de los resultados y no de todo el conjunto de los datos almacenados.[27]

A la hora de hacer peticiones, Cloud Firestore tiene un límite a partir del cual deja de ser un servicio gratuito y empieza a cobrar por número de peticiones. A la hora de hacer el diseño del esquema de modelo de datos tendrá que ser tenido en cuenta para evitar hacer más peticiones de las marcadas por el límite, actualmente en 50.000 peticiones de lectura por día y 20.000 peticiones de escritura y borrado respectivamente.[28]

El siguiente paso, ya conociendo más profundamente la base de datos que se va a utilizar y su funcionamiento, es empezar a averiguar y detallar qué datos o información deberá ser almacenada en la base de datos para la aplicación.

El funcionamiento de la app, resumido y simplificado en información, trata de unir usuarios, conocer la ubicación de éstos y notificar salidas, por lo que se almacenará información sobre los usuarios, las posiciones en las que se encuentran y la configuración que tengan para la alarma.

Después de hacer un análisis del funcionamiento de la aplicación y de la información que va a necesitar, atendiendo a las peticiones CRUD³ que serán necesarias para el buen funcionamiento y desarrollo de la aplicación, además del límite que pueda poner Cloud Firestore en número de peticiones anteriormente mencionado, ha resultado el siguiente modelo de datos:



Figura 15. Modelo de datos
(Fuente propia)

Como primer paso en el modelo de datos, se encuentran las colecciones de Guardianes y de usuarios Protegidos. Dentro de sus respectivos documentos se almacena la información mínima necesaria que necesita la app para su funcionamiento: un email con el que poder hacer el registro y un nombre.

Otra colección es la de Seguimiento, los documentos en ella contendrán la información básica de un emparejamiento entre un Guardián y un Protegido, guardando las identificaciones de ambos tipos de usuarios en la base de datos y un campo booleano para indicar si el seguimiento está activo o inactivo. [RF1]

Siguiendo ahora con las subcolecciones, partiendo de la colección de Protegido se encuentra la subcolección de Última posición, en cuyos documentos se guarda el identificador de la base de

³ Acrónimo para las funciones básicas de creación, lectura, actualizado y borrado en base de datos (en inglés, Create, Read, Update and Delete).

datos del Protegido, así como las coordenadas de latitud y longitud para conocer su posición y un campo fecha en formato Date para tener constancia de cuando se recibió esa última ubicación. [RF6]

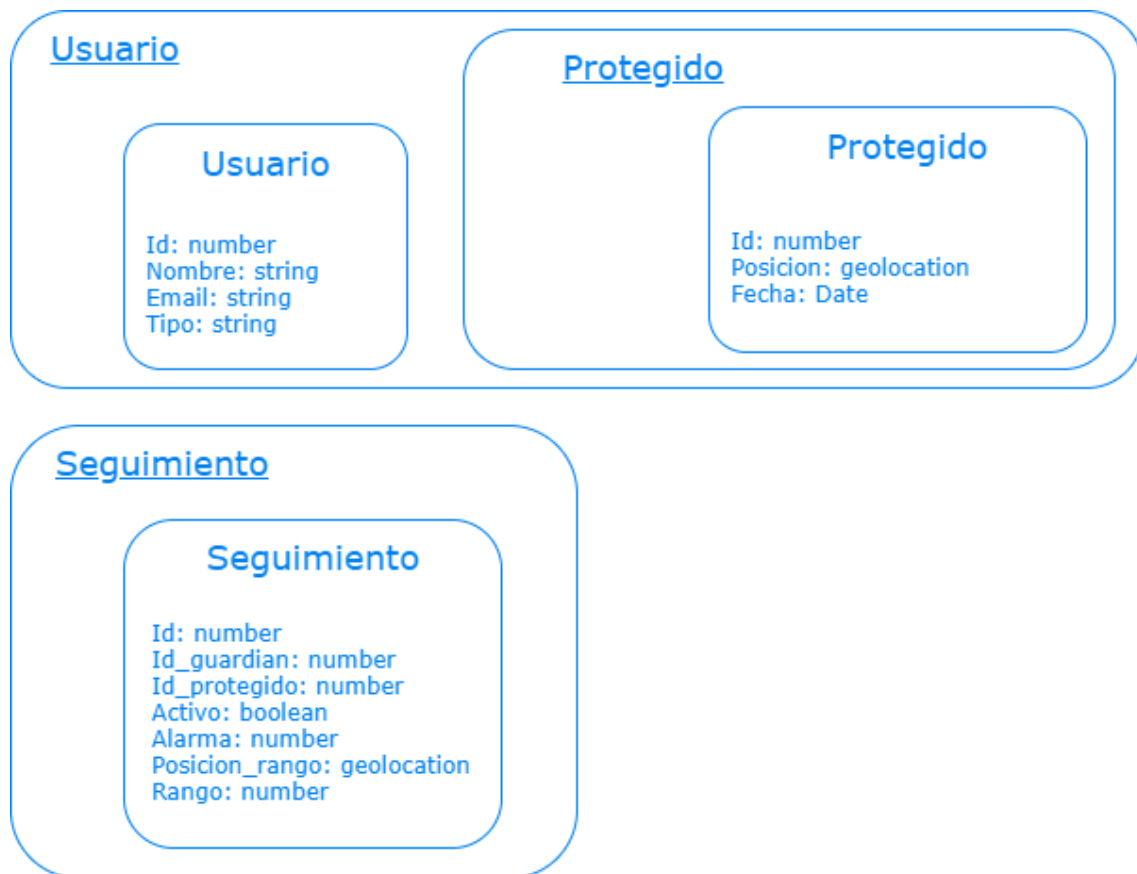
Las subcolecciones de Alarma y Rango están ligadas a Seguimiento. Dentro de Alarma se almacenarán documentos con el identificar del Seguimiento además de dos campos booleanos para conocer si la alarma está conectada y si el sonido de ésta se encuentra activado. [RF4] [RF5]

Por último, la subcolección Rango. En esta subcolección almacena información sobre el rango seleccionado por un Guardián para el seguimiento de un Protegido, en ella se encuentra el identificar del Seguimiento, las coordenadas de latitud y longitud del punto del cual parte el rango circular y el campo rango que contiene la distancia radial de éste. [RF2] [RF3] [RF4]

Hay que tener en cuenta que la mayor parte del contenido almacenado en la base de datos no estará disponible para los usuarios, puesto que será información interna para el funcionamiento o configuraciones determinadas por los propios usuarios, además de que la información personal de estos deberá estar restringida y muy controlado debido a que no se debe poder acceder a información delicada como puede ser la ubicación. Los usuarios únicamente podrán acceder a la última ubicación conocida y cuándo fue recogida, pero solo podrán conocerla cuando la alarma haya saltado.

8.3.2. Avance en el diseño del modelo de datos

Tras analizar de una forma más exhaustiva, se ha decidido rediseñar el modelo de datos del proyecto con el objetivo de reducir el número de colecciones en la base de datos y el número de peticiones que serían necesarias para acceder a datos. En la siguiente figura se muestra el nuevo diseño del modelo de datos ya actualizado.



*Figura 16. Actualización del modelo de datos
(Fuente propia)*

Como primera tabla en el modelo de datos, se encuentran la colección de Usuarios. Esta colección surge de la unión de las anteriores colecciones de Guardianes y Protegidos, puesto que contenían la misma información almacenada dentro de sus documentos: un email con el que poder hacer el registro y un nombre. Además, a causa de esta fusión de colecciones se ha incluido un campo para almacenar el tipo de usuario que es ya sea Guardián o Protegido.

Dentro de la colección principal de Usuarios se encuentra la subcolección de Protegido, que hace referencia a la tabla Última posición del antiguo diseño, en cuyos documentos se guarda el identificador del Protegido, un campo de fecha para saber cuándo se recibió la última ubicación conocida fuera del rango y las coordenadas de latitud y longitud para conocer su última posición, que para este caso Firestore Database tiene modelo de dato denominado punto geográfico que es justo lo que se busca almacenar. [RF6]

Por último, se encuentra la colección de Seguimiento, en la que se ha incluido toda la información que anteriormente se iba a almacenar en las subcolecciones de Alarma y Rango, los

documentos en ella contendrán toda la información necesaria a almacenar de un emparejamiento entre un Guardián y un Protegido.

Se almacenarán los identificadores de ambos usuarios implicados en el enlace, Guardián y Protegido, un campo booleano para conocer si el enlace de seguimiento se encuentra activo o inactivo [RF1] , un campo número para conocer el estado actual de la alarma, siendo cada número entre 0 y 3 un estado entre desconectada, sin sonido, en modo vibración o con sonido como estaría por defecto, [RF4] [RF5] las coordenadas de latitud y longitud como punto geográfico inicial desde el que parte el rango de seguridad y la distancia radial de éste. [RF2] [RF3] [RF4]

8.3.3. Seguridad e integridad

Como se ha mencionado en el párrafo anterior, es importante controlar qué usuarios pueden acceder a qué información y cuándo. La seguridad es un aspecto muy importante de la persistencia para tener en cuenta.

La base de datos Cloud Firestore ofrece la opción de implementar reglas de seguridad para el control de acceso y validación de los datos, en el caso del control de acceso recomienda el uso de Firebase Authentication, el cual se usará en el proyecto con reglas de seguridad de Cloud Firestore. La base de datos comprueba todas las reglas siempre que recibe cualquier solicitud antes de guardar los cambios.[29]

Estas mencionadas reglas de seguridad también limitan y controlan los accesos a colecciones y documentos almacenados. Estas reglas no se heredan, en el caso de que un documento de una colección tenga una subcolección, las reglas de la primera colección no funcionarán para la subcolección, se deberán implementar reglas concretas para la subcolección.

Además, por motivos de seguridad, Cloud Firestore ha fijado unos límites en ciertas acciones o peticiones. La más llamativa y destacada limita a 10 llamadas máximas por solicitud de consulta y 20 para lecturas de distintos documentos.[30]

Por último, Cloud Firestore también avisa enviando alertas en caso de que la base de datos no está protegida correctamente o hubiese algún tipo de vulnerabilidad común en las reglas de seguridad implementadas, así poder arreglar lo que fuese necesario e incluso probar esos cambios antes de implementarlos.

8.4. Diseño Interacción o Experiencia de Usuario

Esta sección estará enfocada y centrada en cómo se busca que el usuario final vea y use la aplicación para que tenga una buena experiencia usándola y no únicamente que esta pueda ser accesible y usable.

Una aplicación puede ser mejor acogida o valorada ya no únicamente porque su funcionamiento sea perfecto y adecuado, si no que uno de los factores que mayor cuenta es la experiencia del usuario. Esta hará sentir al usuario final más cómodo usándola o se le hará más complicado o incluso menos visual o bonito si el diseño de esta es malo, aunque obviamente que el funcionamiento sea correcto es una parte base para que un usuario contemple probar la aplicación y mantener su uso prolongado en el tiempo.

8.4.1. Tipos de personas e intereses

Dado que el diseño de la experiencia del usuario va dirigido a este mismo, al usuario, se va a analizar qué ejemplos de personas podrán usar la aplicación y encajarían como qué tipos de usuarios.

Teniendo en cuenta los Segmentos de Clientes ya analizados en el Lean Canvas del punto 2.2 del trabajo, se mencionará que estos serían encargados o supervisores de personas dependientes, ancianos o menores de edad.

Esto implica que el usuario final de la aplicación debe ser mayor de edad, tener en su propiedad y hacer uso de un dispositivo smartphone y tener a alguien a su cargo o bajo su supervisión, ya sea por motivo profesional o personal, que también tenga a su disposición un dispositivo smartphone. Con esta información se acota, por ejemplo, el público objetivo de la aplicación a adultos entre 18-60 años.

Estos ejemplos de Guardián de la aplicación buscan una aplicación que les avise cuando esas personas a las que siguen salgan de una zona segura, que para cada persona puede ser una casa, un pueblo o un barrio.

Además, pueden necesitar o no interesarles la posición donde se encuentran después de esas salidas, por lo que la aplicación ofrecerá la posibilidad de conocer la última posición almacenada cuando se produzcan dichas salidas.

Por último, habrá situaciones en las cuales la alarma tendrá que estar activada permanentemente y otras en las que no la necesiten, por el mismo hecho de estar presentes con la persona a la que siguen, por lo que la alarma se podrá conectar y desconectar sin limitaciones.

Todo este análisis ha sido contemplado para el análisis y desarrollo de los requisitos, ya que estos parten de las necesidades que tiene el Guardián.

8.4.2. User Journey Map

Conociendo en mayor medida los futuros potenciales usuarios de la aplicación y qué pueden querer buscar en ella, ahora se debe tratar de analizar los pasos que serán necesarios en el uso de Location Noticer y cual debería de ser la experiencia del usuario en cada punto.

Para ello, se va a hacer uso de la técnica denominada “User Journey Maps” en la cual se analiza, visualmente con ayuda de una gráfica, cuál es la emoción que puede sentir el usuario o la reticencia o inseguridad que puede experimentar al hacer cada paso de uso de un servicio, para poder comprender por qué ocurre eso y, en caso de que sea un paso con experiencia negativa, cómo arreglarlo o contrarrestarlo.[31]

En la siguiente gráfica se puede ver el resultado del análisis, en el cual se ha valorado si cada paso de uso en la app ha tenido como respuesta más negativa o más positiva, pudiendo también comparar visualmente.

Los pasos de uso que se han tenido en cuenta para el análisis son la descarga e instalación de la aplicación, el registro recogiendo toda la información necesaria y el inicio de sesión, el enlazado de dispositivos para el seguimiento, la elección o edición del rango de seguridad en un mapa, la posibilidad de configurar la alarma respecto a los valores por defecto, el seguimiento o funcionamiento normal de la aplicación, la recepción de la alarma y la consulta de la última posición almacenada.

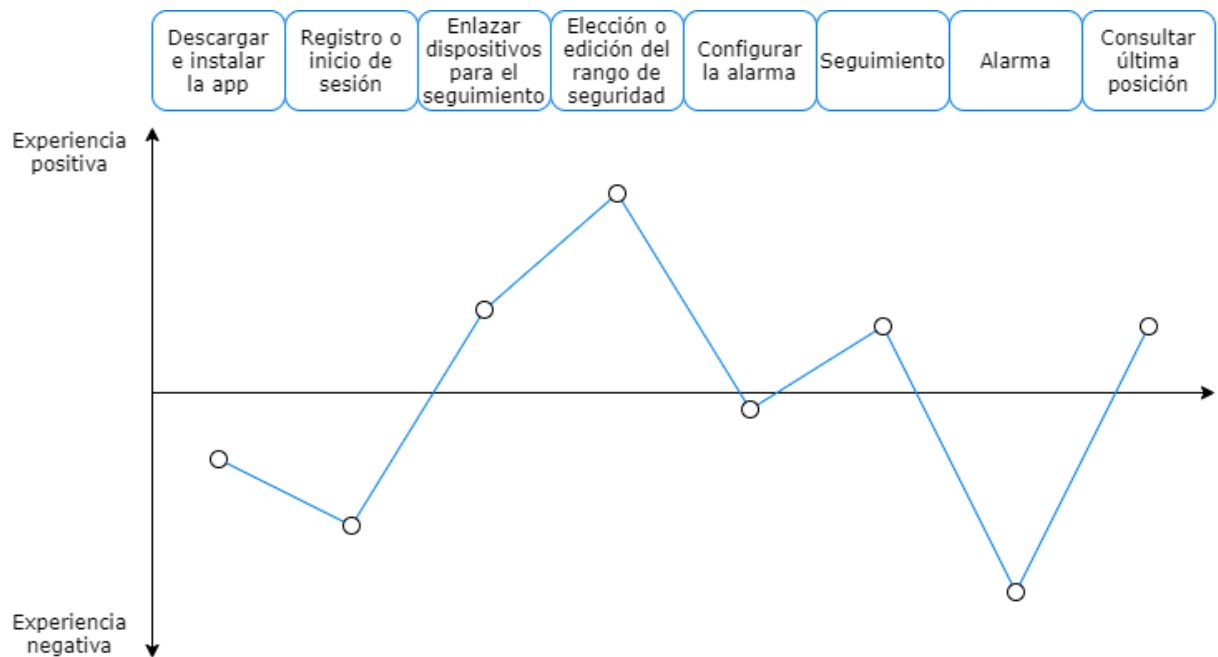


Figura 17. User Journey Map
(Fuente propia)

En el primer paso, como puede ser la descarga e instalación de la app desde una tienda de aplicaciones en smartphones, se valora como una experiencia un poco negativa, debido a lo tedioso que puede ser buscar la aplicación y la espera entre que se descarga e instala. Para este paso poco se puede hacer al respecto más que tratar que el peso de la aplicación no sea muy grande para facilitar la descarga y no ocupar demasiado espacio de almacenamiento.

Después de la descarga e instalación de la app, al ejecutarla por primera vez, lo primero que se debe hacer es registrarse. Una tarea que mayormente provoca pereza, aunque sea necesaria. La acción de registrarse, en este punto de pasos de uso, puede equipararse a la del inicio de sesión en futuras ocasiones en las cuales el registro ya hubiese sido completado.

La experiencia de esta es más negativa puesto que, siendo ya una experiencia negativa, seguida de otra como puede ser la descarga e instalación, da una sensación mayor que por sí sola. Para mejorar este paso, se puede reducir el número de campos a introducir en el registro, únicamente con email y contraseña puesto que el resto de información no es necesaria para el uso de esta, además de guardar la información de inicio de sesión y evitar que la sesión de cierre automáticamente.

El siguiente paso es en el que se enlazan los dispositivos para el seguimiento que, al ser el primer contacto con el contenido real de la aplicación y, además, se trata de una acción tan sencilla

como introducir un código, puede provocar en el usuario su primera experiencia positiva con la aplicación.

A la hora de elegir rango, el usuario ya puede ver la interfaz en la que se muestra el mapa. Es una experiencia positiva puesto que con pocos clicks podrá configurar fácilmente un rango de aviso. Aun siendo una experiencia positiva, puede ser mejorada poniendo como centro del rango por defecto la posición en la que se encuentra el propio usuario, así habría un ahorro de tiempo y de clicks para mayor comodidad del usuario.

La alarma estará activada y tendrá unos valores por defecto, puesto que como es la funcionalidad principal de la aplicación, no tendría sentido que por defecto se iniciara desconectada. El usuario podrá cambiar su configuración libremente, aunque si por sus necesidades tuviese que cambiar esta a menudo, podría ser un tanto molesto. Al tener valores por defecto, esta experiencia podría considerarse más bien neutra, aunque puede mejorar o empeorar según lo que deba cambiar el usuario y con qué frecuencia.

Durante el seguimiento, no es necesario entrar a la aplicación puesto que no se puede ver mucho más que el rango elegido, este paso es simplemente un tiempo de espera a la alarma. Al no ser necesario entrar en la aplicación y esta continúa funcionando, es una experiencia negativa, ya que funciona correctamente sin necesidad de intervención del usuario.

En caso de saltar la alarma, es una reacción en la mayor parte de veces inesperada que, debido al público al que va dirigido, debe ser una experiencia muy negativa dada la incertidumbre y preocupación que provoca. Esta sensación se puede mejorar levemente ofreciendo soluciones u otras funcionalidades que puedan servir complementarias a la situación de forma inmediata.

Otras funcionalidades como la consulta de la última posición conocida que ha sido almacenada en base de datos. Puesto que este paso va ligado al anterior y sirve para mejorar el impacto negativo, su experiencia será ligeramente positiva, ya que se podrá conocer la posición del Protegido.

Como conclusiones, se deberá tener en cuenta en reducir al mínimo posible el tiempo en el que transcurren pasos de instalación o configuración que puede ser más tediosos.

Además, es muy importante tener mucha claridad y mostrar de una manera directa la alarma y la última posición, puesto que puede tratarse de situaciones delicadas y que implican un tiempo de respuesta muy rápido, ahorrar cualquier intermediario o impedimento al usuario en estas acciones tendrá un impacto positivo en la experiencia de este.

8.5. Guía de estilos

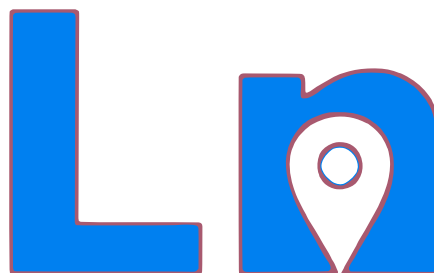
En este apartado del diseño se definirán los estilos visuales o decisiones estéticas, como pueden ser colores corporativos, logotipo e icono de la aplicación o tipo de fuente, a seguir durante el proceso de diseño y de implementación de la aplicación.

8.5.1. Logotipo e icono

La aplicación o producto final saldrá a las tiendas de aplicaciones con el nombre de Location Noticer, que puede traducirse del inglés al castellano como Notificador de ubicación. Título que describe brevemente el propósito y funcionalidad del proyecto.

Conociendo el nombre de la aplicación y su temática, se ha diseñado un icono y un logotipo de acorde a ello. Además, también han influido en el diseño detalles que serán justificados en los apartados venideros de la Guía de Estilos como los colores corporativos.

El icono está formado por las iniciales de Location noticer, Ln, en color azul con bordes redondeados en un color rubí. Como añadido visual, el espacio interior que contiene la letra n simula ser la iconográfica que marca una ubicación.



*Figura 18. Icono Location noticer
(Fuente propia)*

Para la marca, el color azul aporta frescura, tranquilidad y confianza, atributos de gran valor para los Segmentos de Clientes del proyecto. Además, sus bordes y esquinas están redondeados para mostrar y transmitir modernismo, así como ligereza y fiabilidad.

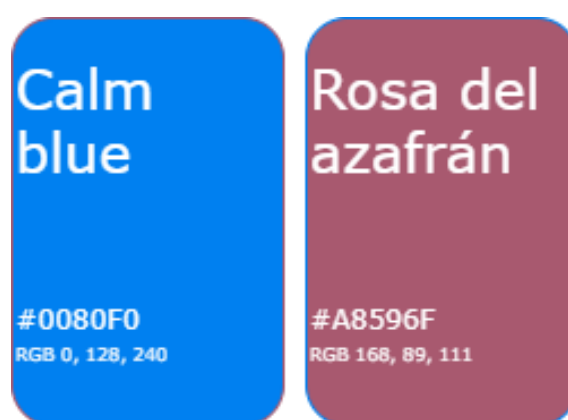
Conocido ya el icono de la app, se diseña de una forma simple y partiendo de este el logotipo. Este está formado por el simple nombre de la aplicación, escrito también en Verdana, pero utilizando las letras Ln del icono en lugar de las que tendría las propias palabras escritas.

Location oticer

*Figura 19. Logotipo Location noticer
(Fuente propia)*

8.5.2. Colores corporativos

Destinados para el diseño de la marca como el de la aplicación, se han seleccionado dos colores fríos, como son el azul y un color frío cercano al rubí.



*Figura 20. Colores corporativos
(Fuente propia)*

De estos dos colores, denominados Calm blue y Rosa del azafrán, serán considerados colores principal y secundario respectivamente. Además, se utilizará el blanco como tercer color o alternativo para su uso como base o fondo en interfaces o para color de texto con el objetivo de mejorar la comprensión, lectura y visualización de interfaces en el uso de este combinado con los colores corporativos.

8.5.3. Fuente y tipografía

La fuente elegida para ser utilizada en la aplicación es KoHo, una fuente de Google Fonts diseñada por Cadson Demak. KoHo es una fuente sans serif ⁴, que aporta sensación de ser tipografías más modernas y se caracterizan por ser más legibles, lo cual es una ventaja para su uso en la aplicación. Además, está sugerida por expertos para el uso en pantallas.[32]

KoHo es una familia de fuentes tailandesa y latina, diseñadas para no parecer ni demasiado mecánicas ni demasiado caligráficas, y obteniendo como resultado una fuente recomendada tanto para texto escrito como para fines de visualización.

En la siguiente figura se puede visualizar un ejemplo de la fuente KoHo aplicándole los colores corporativos.

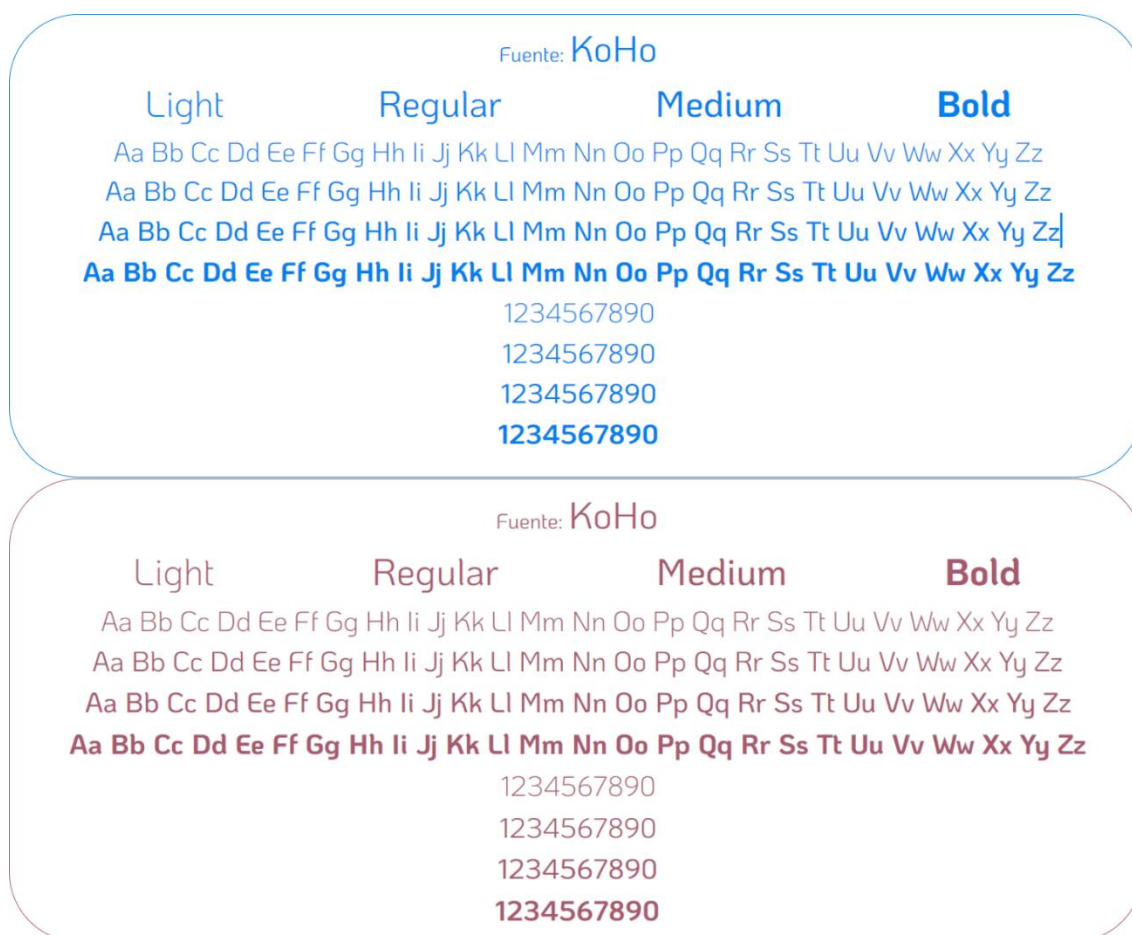


Figura 21. Fuente KoHo
(Fuente propia)

⁴ Sans serif o palo seco es un tipo de letra en tipografía sin pequeñas terminaciones como son remates o serifas.

Esta fuente pone a disposición el usuario distintos estilos como son Extra-light, Light, Regular, Medium, Semi-bold y Bold, además de la versión en Italic de todas las mencionadas. Aunque, de todas estas, se han seleccionada únicamente los estilos Light, Regular, Medium y Bold, ya que no se consideran necesarios el resto de estilos o que vayan a poder ser utilizados en la aplicación o en el diseño de ésta.

8.6. Diseño Interfaces

Este apartado se centrará en el diseño visual de la aplicación, mostrando una imagen lo más aproximada posible de cómo será la aplicación en su estado final. Además, al realizar el diseño, tendrán que verse reflejados todos los requisitos RF y RNF definidos en el apartado 7.2 Requisitos.

La finalidad de este apartado también es ayudar a reducir el tiempo que se va a emplear durante la implementación, debido a que se busca tomar el mayor número de decisiones posibles en este momento y obtener una gran fidelidad con el producto final, así en el momento de implementar será posible dedicar todo el tiempo únicamente a eso, a escribir el código de lo ya diseñado.

Como primer paso, se hicieron unos primeros bocetos de wireframes a papel y lápiz, para tener una primera vista de cómo se pretendía que quedase. Un wireframe es una de las primeras propuestas de solución, un diseño de baja fidelidad que orienta sobre qué funcionalidades tendrá la aplicación y donde estarían situadas en la interfaz.

En la siguiente figura se muestra el diseño de los wireframes iniciales que fueron pensados para Location noticer. En ellos se pueden ver el funcionamiento de la aplicación con intención de mostrarlo en orden cronológico por cómo está pensaba para ser usada.



Figura 22. Wireframes de Location Noticer
(Fuente propia)

En la figura se puede observar que están bocetadas las interfaces iniciales, como las de enlazado de dispositivos con el código, selección de rango, seguimiento, alarma y configuración.

Ya en estos wireframes se tienen en cuenta varios requisitos y funcionalidades: se pueden enlazar dispositivos [RF1], ver el mapa y seleccionar el rango en él [RF2][RF3] y recibir y visualizar la alarma y poder cancelarla o silenciarla [RF4][RF5].

Y, tras los wireframes, el siguiente paso tomado en el diseño de las interfaces ha sido la realización de mockups, una representación visual con mayor fidelidad y detalles estéticos. Los diseños de las interfaces en mockups deben ser prácticamente idénticos al que será el resultado

final de la aplicación, o al menos la finalidad es guiarse con los mockups en la implementación del diseño final.

En los mockups han sido tenidos en cuenta y aplicados todos los detalles mencionados en la guía de estilos como pueden ser fuente y tipografía, colores corporativos y logotipo e icono. Además, en ella también se verán los elementos situados en la interfaz en su lugar y con su tamaño ideal.

Para la realización de estos mockups de alta fidelidad se ha utilizado la herramienta Figma online, que permite el diseño gratuito de un proyecto.[33]

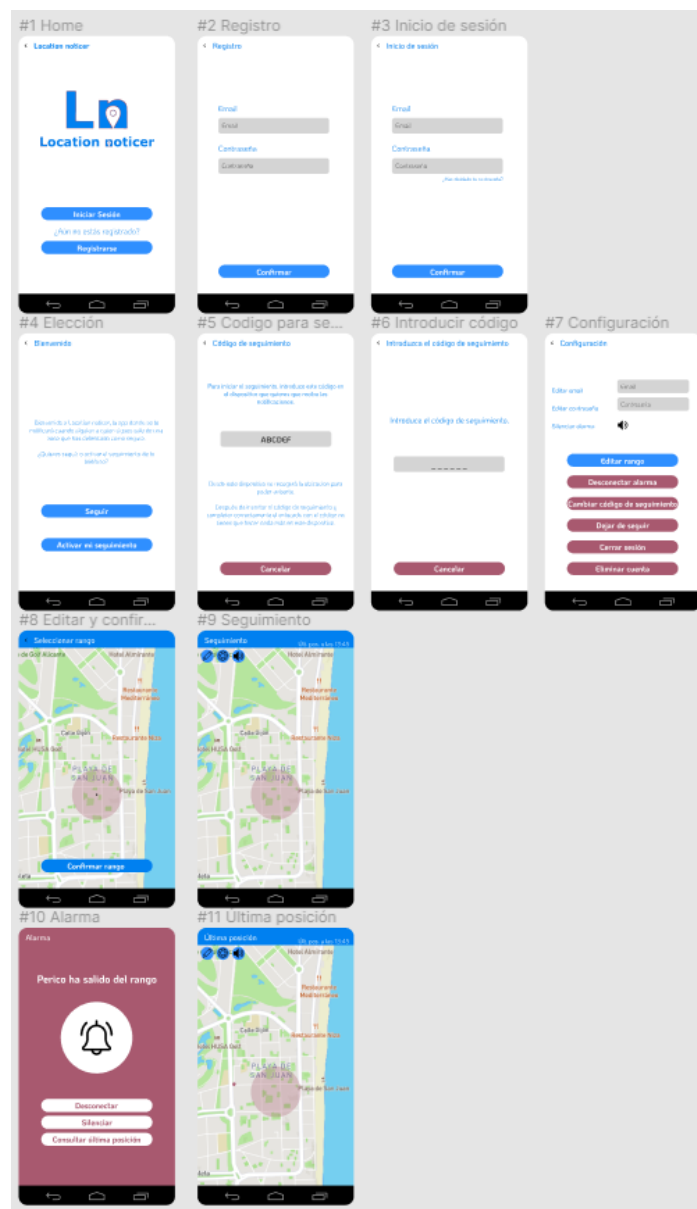
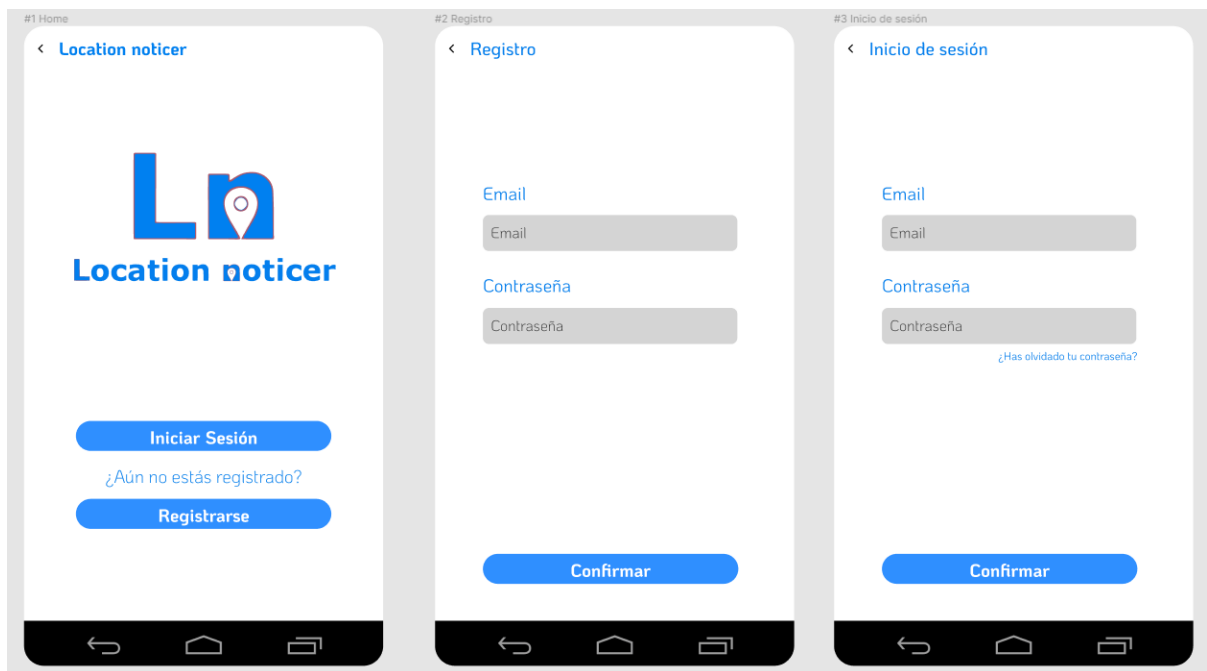


Figura 23. Todos los mockups desde Figma
(Fuente propia)

En la figura anterior, se muestran todas las interfaces diseñadas, un total de 11 ampliando las 8 resultantes de los wireframes, a las cuales se han sumado la consulta de la última posición [RF6], las de inicio de sesión y registro.



*Figura 24. Interfaces 1, 2 y 3
(Fuente propia)*

Las primeras 3 interfaces son las primeras que se encontrará cualquier usuario al abrir la aplicación en caso de estar recién instalada. La primera es una interfaz de bienvenida, en la que se puede visualizar el logotipo e icono de la app y las opciones de iniciar sesión o registrarse. Las siguientes son muy similares, puesto que son la interfaz de inicio de sesión y de registro, cuya única diferencia es la opción de recuperar/cambiar contraseña en el inicio de sesión por si se hubiese perdido.

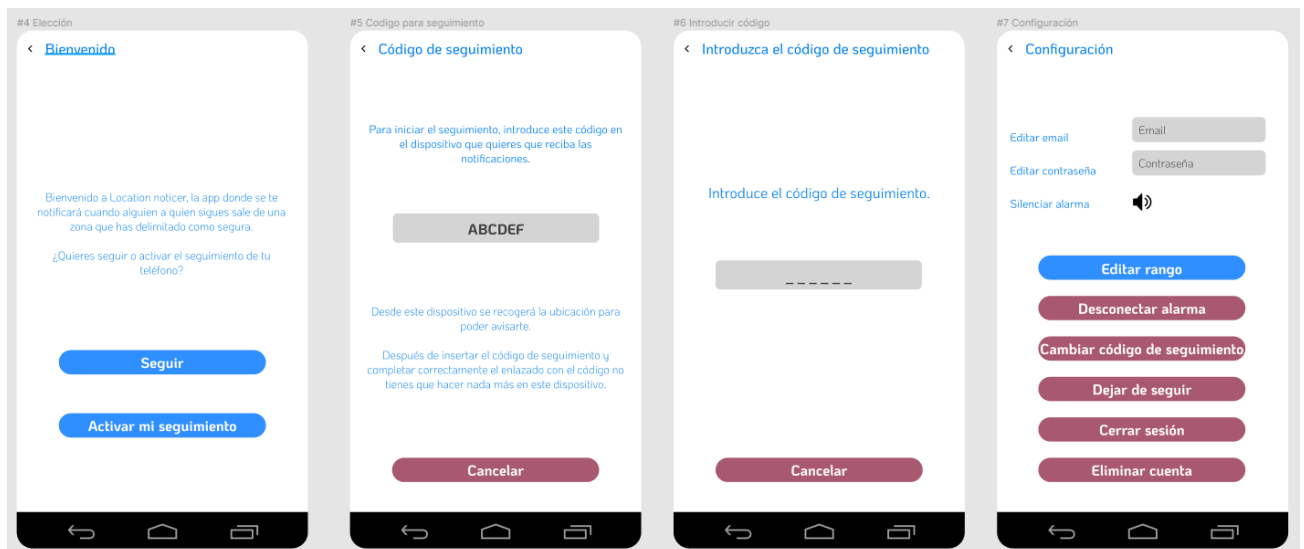


Figura 25. Interfaces 4, 5, 6 y 7
(Fuente propia)

Tras el registro en la aplicación, se deberá elegir si la finalidad de la cuenta va a ser hacer un seguimiento o emitir su posición para que haga otro usuario el seguimiento, es decir, en esta interfaz se decide si el usuario va a ser de tipo Guardián o Protegido.

En caso de ser Protegido, se mostrará una interfaz con un código generado que deberá insertarse en la interfaz del Guardián para completar el enlazado de dispositivos para el seguimiento.

Por último, aunque esta interfaz no tenga por qué estar incluida en este orden, se encuentra la interfaz de configuración, a la cual se accede desde un enlace en interfaces posteriores, desde la que se puede editar la información del perfil de usuario además de silenciar o desconectar la alarma, editar el rango de aviso, cambiar o desconectar el seguimiento, cerrar sesión o eliminar la cuenta.

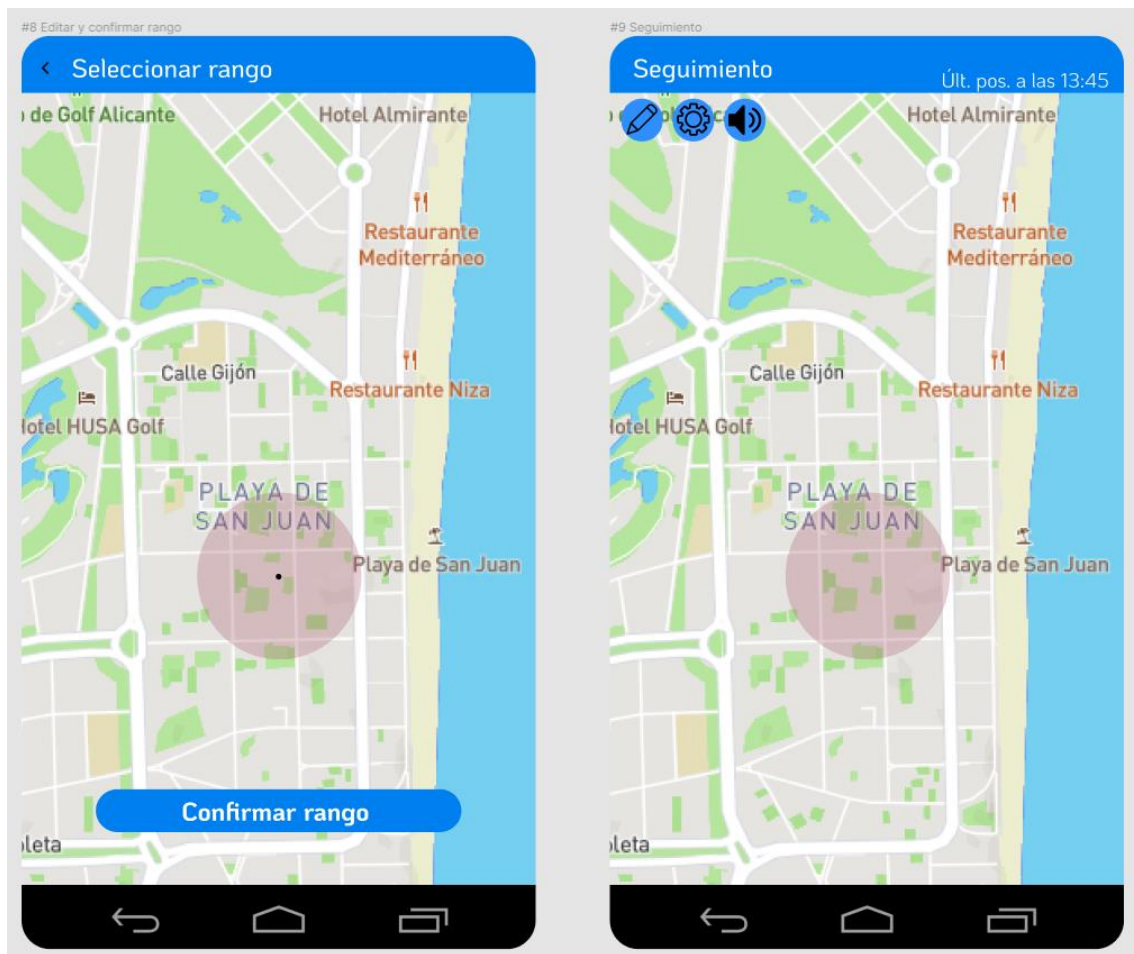


Figura 26. Interfaces 8 y 9
(Fuente propia)

Las siguientes dos interfaces son las primeras de la aplicación en las que aparece un mapa. En la primera se deberá seleccionar el rango de forma táctil pulsando y ampliando en el mapa, para finalizar el enlace de seguimiento o para editarlo cuando ya estaba enlazado.

La segunda, más general, es una interfaz de seguimiento, aunque realmente no hace ningún seguimiento visual puesto que no dibuja la posición de dónde se encuentra el usuario Protegido, aunque sí que sirve para comprobar que no ha salido del rango seleccionado, ver dónde está situado este y con botones para poder editar el rango, silenciar la alarma o acceder a la interfaz de configuración. Además, en esta interfaz se podrá ver la hora guardada de cuándo fue la última vez que el Protegido estuvo fuera de la zona delimitada.

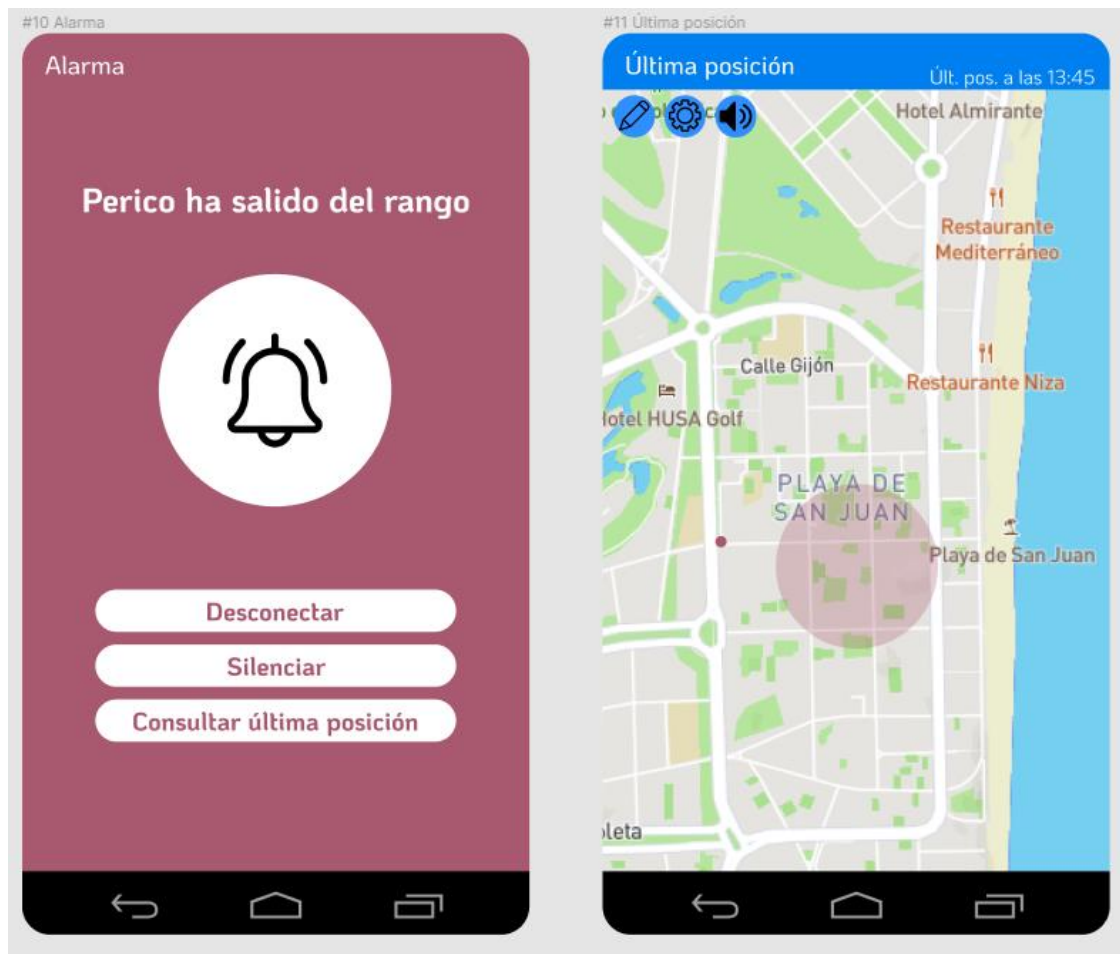


Figura 27. Interfaces 10 y 11
(Fuente propia)

Las últimas interfaces a las que tendrá acceso un usuario Guardián son las de alarma y última posición, las cuales solo son accesibles si la alarma ha saltado cuando Protegido sale del rango delimitado.

En la interfaz de alarma, como se puede observar, se podrá desconectar y silenciar la alarma y, además, se podrá acceder a la interfaz en la que se mostrará la última posición. Desconectar la alarma implicará que ya no se notificará más ninguna salida hasta que se vuelva a activar, y silenciar implica que se notificará, pero sin emitir sonido o vibración alguna. Esta interfaz es idéntica a la interfaz de seguimiento, con la diferencia de que sí que señala la ubicación exacta de dónde se encuentra Protegido.

Con todas las relaciones entre interfaces ya mencionadas, se ha generado el siguiente mapa de interfaces de la aplicación, desde el cual se puede ver de qué interfaces se puede ir a cuáles otras.

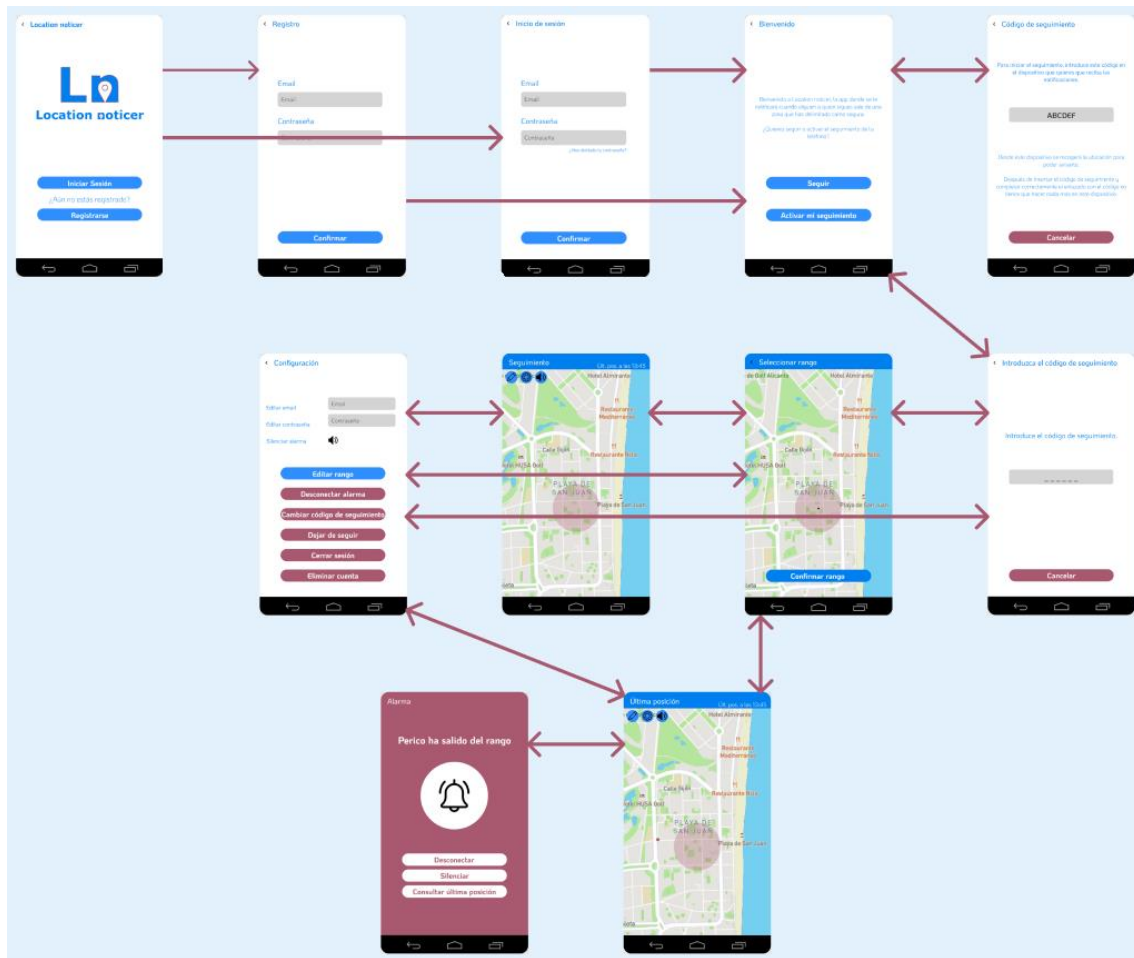


Figura 28. Mapa de interacción de interfaces
(Fuente propia)

8.7. Avance en el diseño de interfaces

Tras un primer apartado de Diseño de Interfaces, se ha decidido introducir un nuevo apartado con los avances tras esos diseños. En ese primer apartado se presentan los primeros diseños y en este, los cambios que han sido realizados respecto a los primeros y las decisiones o motivos por los cuales se han dado. Ya que el diseño es evolutivo, es enriquecedor conocer la evolución de las interfaces en este proceso.

Dado que ya se han presentado wireframes e incluso unos primeros mockups, en este apartado se presentarán directamente unos mockups finales más refinados con lo que podrá ser la aplicación resultada de la implementación.

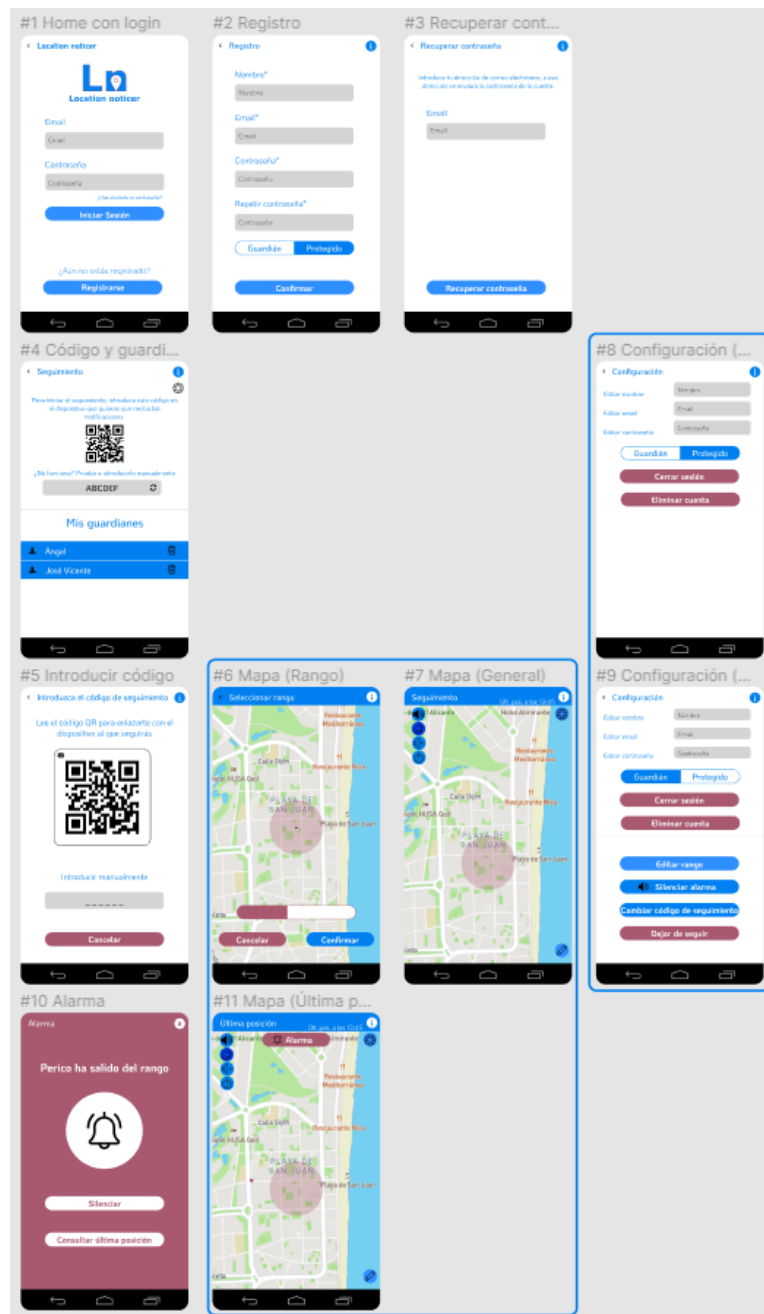
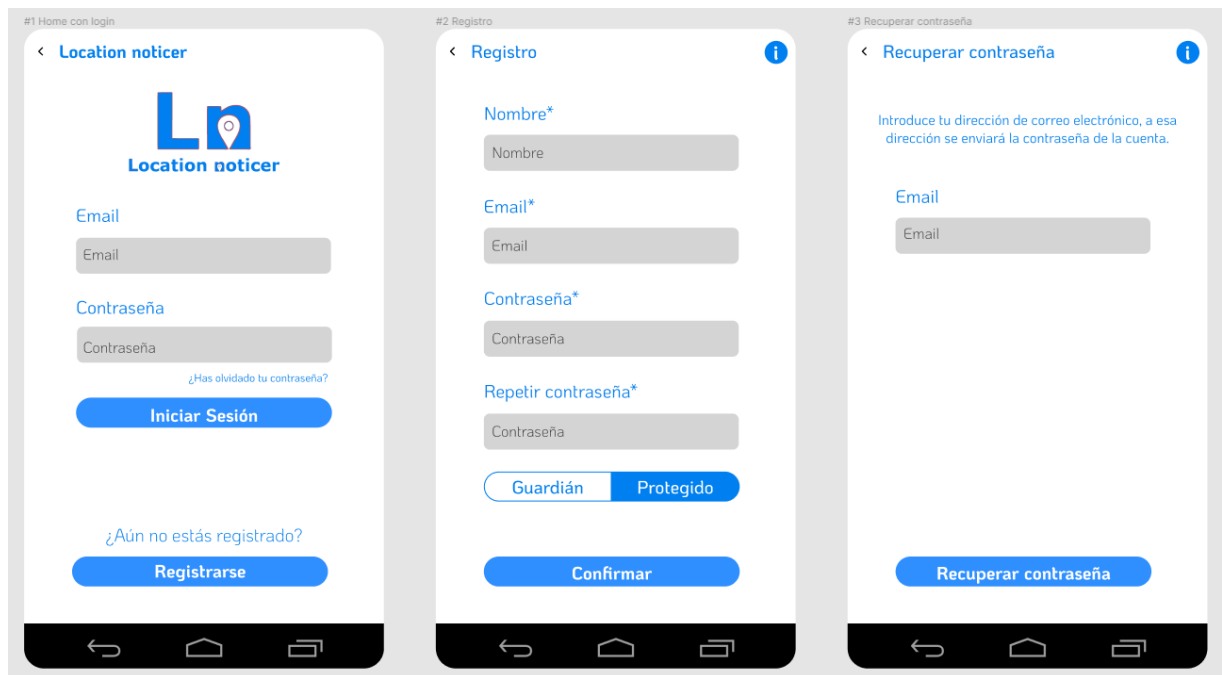


Figura 29. Mockups finales
(Fuente propia)

Como ocurre en el anterior punto, primero se puede observar una imagen general de todas las interfaces de la aplicación en la herramienta online Figma.

A diferencia de la anterior, en esta se puede observar unos recuadros azules que abarcan distintas interfaces. Esto significa que esas interfaces serán la misma, simplemente han sido diseñadas más de una vez para poder conocer cuáles serán sus diferentes estados mientras sean usadas.



*Figura 30. Interfaces 1, 2 y 3
(Fuente propia)*

Inciendiendo más en cada interfaz individualmente, se pueden observar estas tres interfaces en primer lugar. A diferencia de las anteriores, en esta actualización la interfaz de inicio de sesión o login está conjunta a la página principal, por lo que será posible iniciar sesión nada más iniciar la aplicación ahorrando de esta manera una interacción al usuario respecto al diseño anterior.

Además, las interfaces de registro y elección también han sido diseñadas conjuntas esta vez, por lo que se ahorra una interfaz, la cual simplemente amplificaba la información que debía ser recogida en el mismo registro.

La última de estas tres interfaces se corresponde a la de recuperación de contraseña, una interfaz no contemplada en el diseño anterior. A esta interfaz se accederá si se desea recuperar esta misma en el caso de que no se recuerde la contraseña en el inicio de sesión.



*Figura 31. Interfaz 4
(Fuente propia)*

La interfaz 4 corresponde a la interfaz que contenía el código de seguimiento del anterior diseño que solo podía visualizar un usuario Protegido. En este caso, se ha tenido en cuenta que el enlazado se pueda realizar con un código QR y, en caso de que no funcionase, poder introducir manualmente el código inicial.

Además, se ha introducido un apartado en el que se muestra una lista de Guardianes, de esta forma el usuario Protegido podrá observar todos los Guardianes a los cuales se encuentra enlazado y con la posibilidad de eliminarlos libremente. Gracias a este añadido en la interfaz, esta podrá ser usada como interfaz general para los usuarios Protegidos después de hacer el inicio de sesión si ya están registrados.

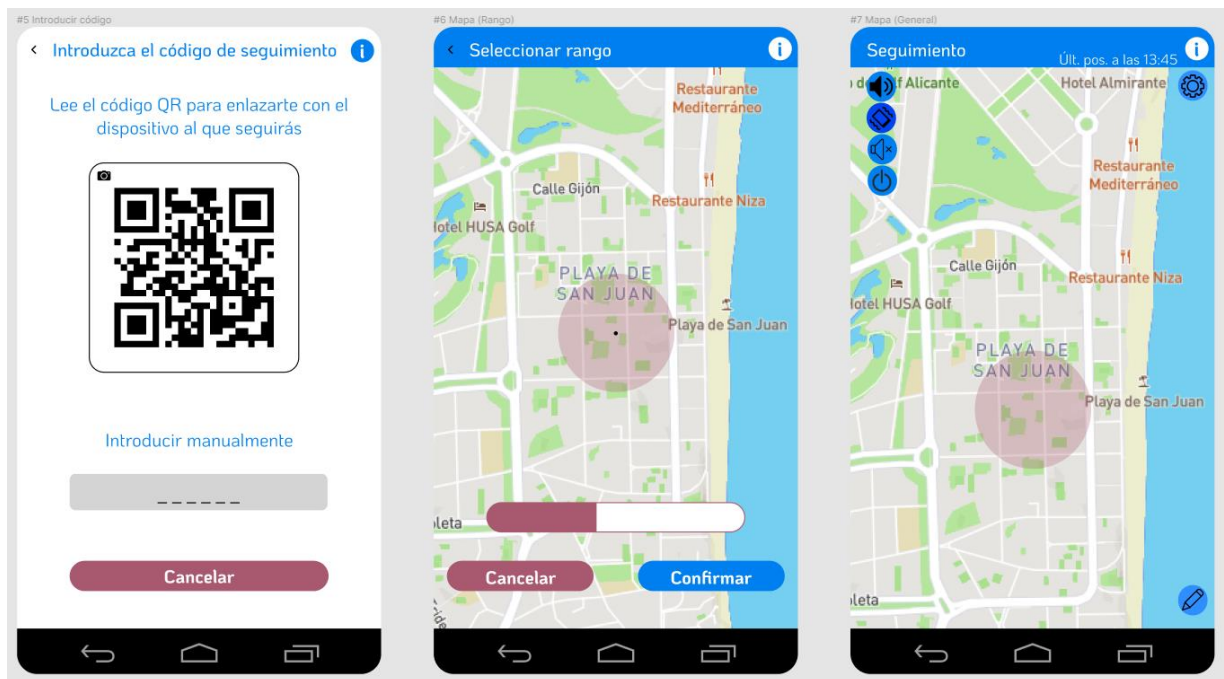
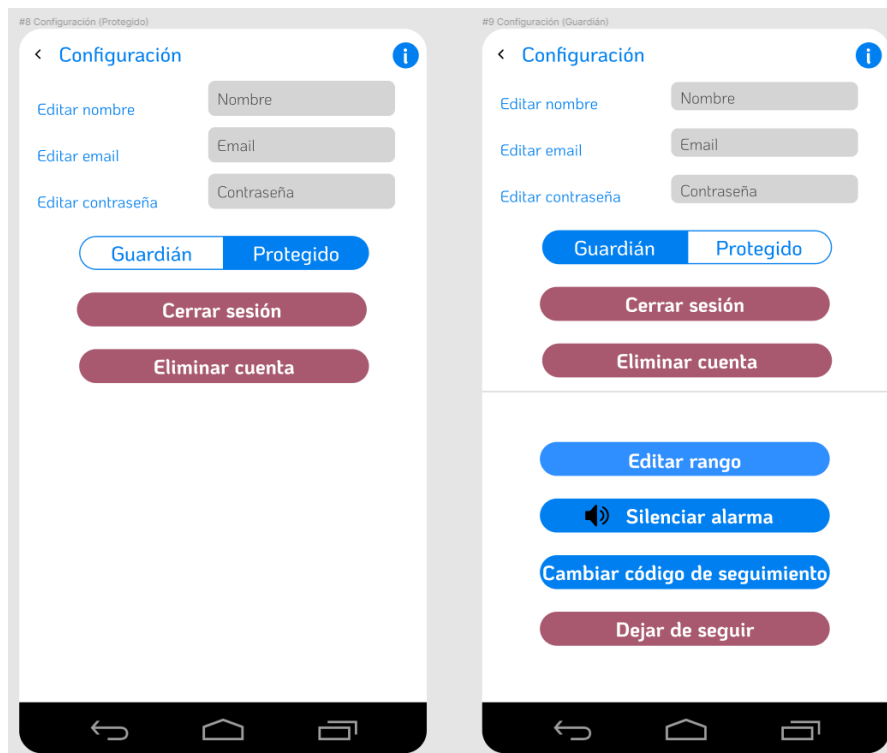


Figura 32. Interfaces 5, 6 y 7
(Fuente propia)

La primera de este grupo de interfaces servirá para leer el código QR de enlazado o, en su defecto, introducir el código de seguimiento correspondiente. La diferencia con su diseño anterior es la incorporación de la lectura del QR.

En el caso de la interfaz de selección de rango sería la primera en la cual se visualiza el mapa. En ella se puede seleccionar el punto inicial y el rango desde los cuales se informará de salidas a los Guardianes. A diferencia del diseño anterior, el rango se podrá elegir desde una barra progresiva y no de forma táctil en el mapa como inicialmente estaba pensado. Además, también se podrá visualizar, siempre que no sea la primera vez que se selecciona el rango de seguridad, un botón de cancelar.

Por último, la interfaz de seguimiento no ha sufrido grandes cambios. Sigue siendo una interfaz de espera a posibles alarmas o de acceso a editar la configuración, el rango o estado de sonido o activación de la alarma. Concretamente esto último se encuentra en la misma zona con cuatro botones con los cuatro estados: activa, vibración, silenciosa o apagada.



*Figura 33. Interfaces 8 y 9
(Fuente propia)*

Las interfaces 8 y 9 se corresponden a una misma interfaz de configuración, pero visualizada como usuario Guardián y como Protegido, como es obvio la interfaz a la que puede acceder Guardián dispone un apartado para configurar la alarma que no tendrá un usuario Protegido.

Como nueva funcionalidad se ha introducido la posibilidad de cambiar de rol entre Guardián y Protegido. Además, se han juntado las opciones de silenciar y desconectar alarma, pudiendo seleccionar entre alarma en silencio, en vibración, con sonido o desconectada.

Además, como mejoría visual, esta interfaz ahora está dividida en dos: la parte superior mostrará la configuración correspondiente al perfil de usuario y la parte inferior, que estará solo disponible para Guardianes como ya se ha mencionado, se mostrará la correspondiente a la alarma.

Los botones de color más cálido indican que son funcionalidades más restrictivas o negativas respecto a las azules, que son simplemente cambios que no implican eliminar el seguimiento, la cuenta o la aplicación.

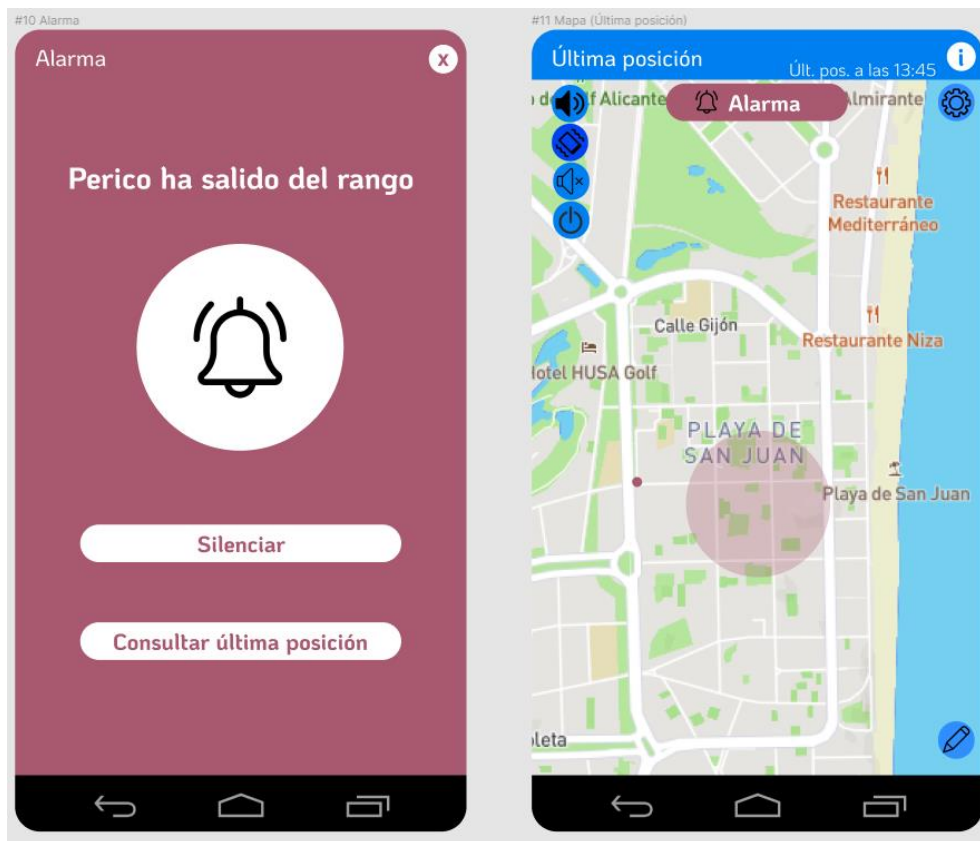


Figura 34. Interfaces 10 y 11
(Fuente propia)

La interfaz 10 corresponde a la interfaz de la alarma, la cual saltará automáticamente cuando el usuario Protegido salga del rango de seguridad seleccionado. Como se ha mencionado en la explicación de la interfaz 9, se elimina el botón de desactivar alarma ya que la opción estará incluida en el de silenciar. Además, se incluye un botón de cerrar para poder salir de nuevo a la pantalla principal o salir de aplicación.

La página única de última posición queda eliminada y será implementada conjunta a la interfaz 7 de seguimiento, incluyendo que en el caso de estar la alarma activada se podrá visualizar la última posición almacenada en base de datos y un pequeño aviso visual de que la alarma ha saltado o sigue activa.

Por último, una funcionalidad añadida que tienen en común prácticamente todas las interfaces es el botón de información o ayuda, desde el cual se desplegará un mensaje personalizado para cada interfaz explicando su contenido y funcionamiento por si fuese necesario para el correcto uso de la aplicación.

En este caso no se ha rediseñado un mapa de interacción para las nuevas interfaces, puesto que son las mismas conexiones que en el mapa del punto anterior salvo con pequeños cambios que no interfieren en este.

8.8. Diseño de pruebas y validación

En este apartado se definirán una serie de pruebas o acciones a tomar para la comprobación del correcto funcionamiento de la aplicación. Y no sólo eso, sino que también se deberá comprobar y validar que los requisitos definidos, ya sean RF o RNF, han sido cumplidos.

Para todo ello, además de conocer mejor el funcionamiento del proyecto para poder mejorarlo, se ha de elegir qué tipo de pruebas serán las necesarias para ello, también teniendo en cuenta que puede implicar una gran cantidad de tiempo del que se puede no disponer y no tratando el TFG únicamente sobre ello.

Firebase, aprovechando que va a ser utilizado como hosting, base de datos, autenticación de usuarios e incluso formará parte de la implementación, también ofrece servicios para realización de pruebas de calidad a tener en cuenta y que serán utilizados para las pruebas y validación de este proyecto.

Se utilizará Firebase Crashlytics [34] para tener informes y reportes en tiempo real de los errores que pueda dar la aplicación. Ayuda a priorizar qué errores pueden ser más importantes que otros para los usuarios. El uso de esta herramienta estará enfocado únicamente a Android a pesar de también poder aplicarse a iOS.

Como herramienta para la monitorización del rendimiento o la latencia de los usuarios en tiempo real al utilizar la aplicación servirá Firebase Performance Monitoring [35]. Además, esta herramienta facilita comprobar el funcionamiento cuando se incluyen nuevas funcionalidades o se ejecutan cambios en la configuración. Además, conocer mejor el contexto donde ocurren los problemas ayudará a reducir el tiempo de resolución de estos.

Por último, para la ejecución de pruebas servirá Firebase Test Lab [36]. Esta herramienta permite hacer pruebas sin necesidad de que sean con tus usuarios reales, ejecuta la app en dispositivos físicos y virtuales que simulan entornos de uso reales.

Además, dispone de un navegador automático, Robo, que navega por la app con la finalidad de generar informes de errores incluso con capturas de pantallas con el fin de solucionar los problemas antes de que los encuentre el usuario final.

9. Implementación

Tras haber definido en los puntos anteriores una metodología con la que trabajar, definido unos requisitos con los que posteriormente se diseñó la aplicación, en este apartado se explicará y redactará el proceso de implementación de la aplicación.

Tal como se mencionó en la metodología, la implementación estará dividida en sprints. Estos sprints constarán de objetivos y su tiempo será variable y no estará planificado previamente.

La implementación implica el proceso de llevar a cabo la aplicación, ya sea adecuando el entorno donde se va a desarrollar o escribiendo el código. En este punto de la memoria se explicarán todos los detalles de los sprints, incluso incluyendo los problemas que puedan surgir y que decisiones se toman sobre la marcha si es que es necesario hacerlo.

El objetivo personal marcado en la implementación es, además de conseguir un mínimo producto viable y avanzar desde ese punto lo máximo posible a parecerse idénticamente al diseño, poder aprender a hacer todos los detalles de la gestión e implementación de una aplicación de este calibre.

9.1. Sprint 1: Preparación y configuración del proyecto

Para poder iniciar la implementación de una aplicación, se debe adecuar su entorno previamente y configurar el proyecto correctamente. En eso se ha basado este sprint, en configurar todo el entorno del proyecto y dejarlo preparado para la implementación.

En primer lugar, se ha elegido Visual Studio Code [37] para emplearlo como IDE⁵ en el desarrollo del código del proyecto, un editor de código desarrollado por Microsoft. Se ha seleccionado Github [38] como repositorio para almacenar el código y poder llevar un control de versiones. A pesar de que Github se puede usar desde su propia interfaz web, se ha instalado Github Desktop la para mayor facilidad a la hora de realizar las cargas y descargas de código. Además, también se utilizará la herramienta Source Control de Visual Studio Code conectada con el repositorio del proyecto de Github.

⁵ IDE: Entorno de desarrollo integrado (Integrated Development Environment en inglés). Programa informático que ofrece servicios para el diseño de aplicaciones, Incluyendo editor de código, compilador y depurador.

Tras instalar todo lo mencionado hasta el momento, se creó el proyecto de Ionic donde se programará todo y que será la aplicación, la plantilla por defecto utilizada será “blank” sin ningún tipo de menú ni contenido por defecto. Ionic [39] es un framework basado en Angular [40] que también requiere el uso de TypeScript [41] para el desarrollo de código en él. Ionic será ejecutado por Visual Studio Code en un servidor localhost abierto por el propio Ionic, al cual se accederá desde el navegador por defecto, en este caso el navegador a utilizar será Google Chrome [42]. Con el objetivo de comprobar mejor el funcionamiento en dispositivos reales, también se contará con un smartphone Samsung Galaxy A50 para ejecutar la aplicación en él.

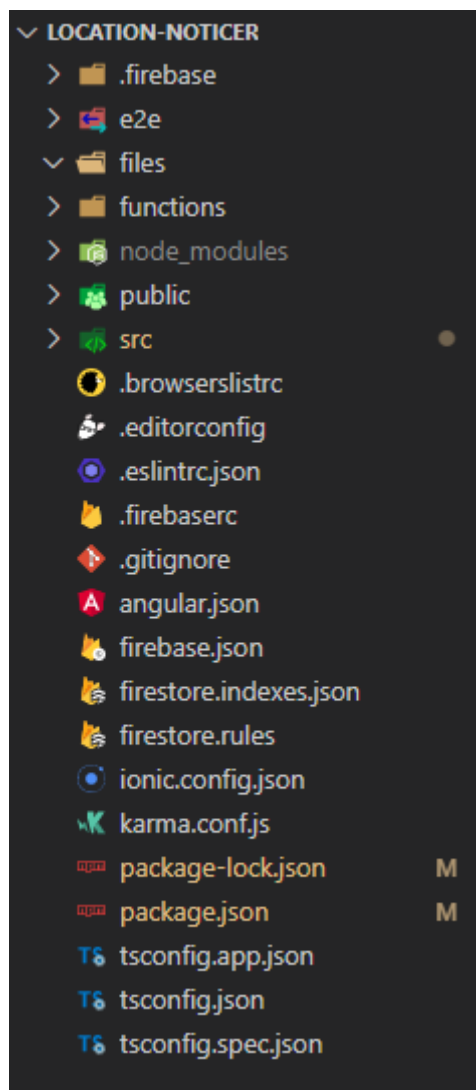
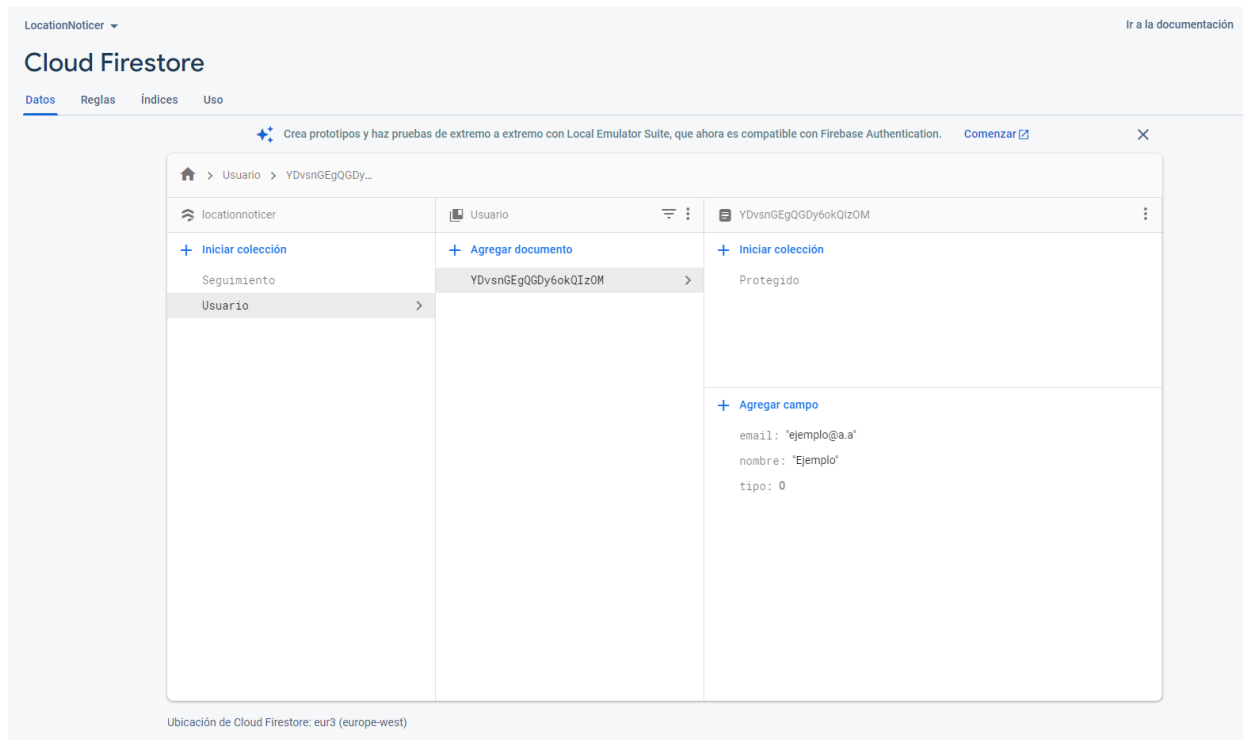


Figura 35. Directorio del proyecto de Ionic en Visual Studio Code
(Fuente propia)

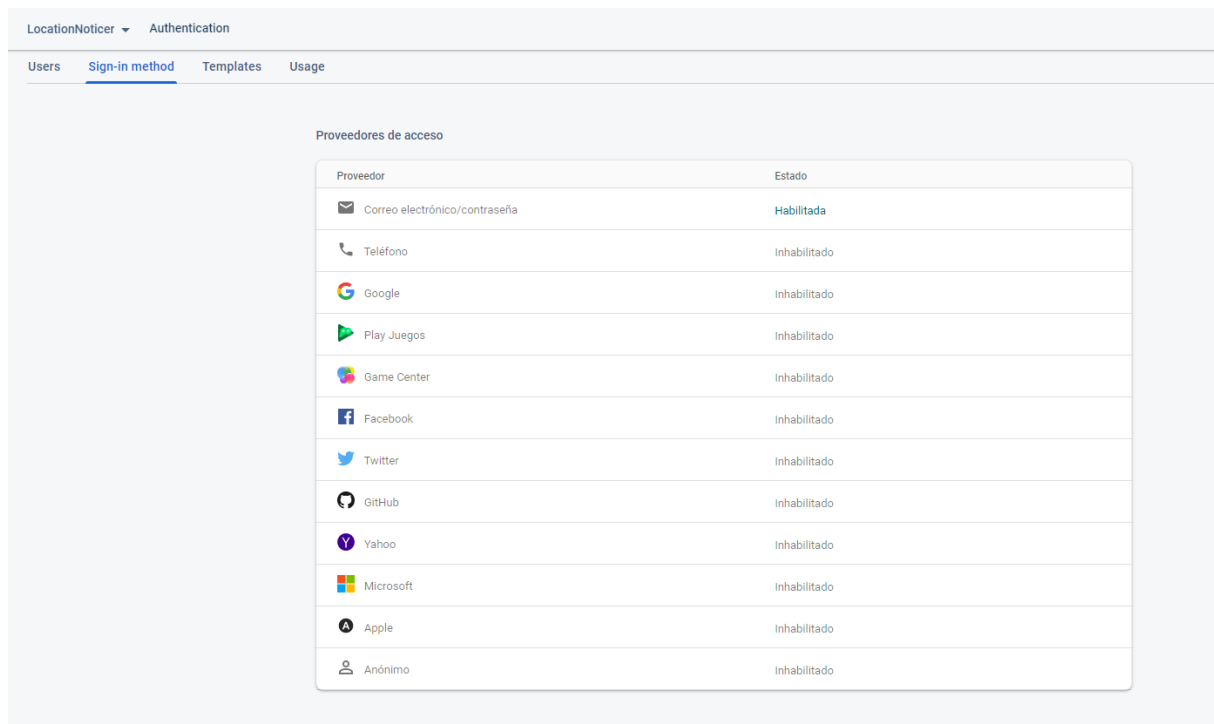
Después de la generación del proyecto en Ionic, se han instalado librerías necesarias para la aplicación. En primer lugar, para poder conectar con Firebase y su base de datos (Firestore Database) y gestión de usuarios (Authentication) se ha instalado la librería AngularFire [43], una

librería oficial de Angular dirigida a lo que se busca, conectar con Firebase. Otra librería que ha sido instalada para su futuro uso es Mapbox [44], para poder insertar e implementar mapas en la aplicación más adelante. Con el objetivo de facilitar la implementación con elementos nativos de smartphones, también se ha instalado la librería de Cordova [45]. Por último, también se ha instalado el framework Bootstrap [46] para facilitar la edición de los estilos de las interfaces de la aplicación.



*Figura 36. Firestore Database configurado con ejemplos introducidos
(Fuente propia)*

Además, se ha creado e iniciado un proyecto nuevo en Firebase configurando con ayuda de su documentación los servicios de Firestore Database, creando la base de datos siguiendo el modelo de datos diseñado previamente, y de Authentication, habilitando el registro de usuarios en la aplicación con correo y contraseña entre las diversas opciones que permite.



*Figura 37. Firebase Authentication habilitada con email y contraseña
(Fuente propia)*

En resumen, este sprint ha estado centrado en la instalación y configuración de todo lo que será necesario para implementar el proyecto, incluyendo también gran parte de investigación sobre las tecnologías que se ha decidido utilizar y haciendo pruebas en ellas.

9.2. Sprint 2: Primeras interfaces y la gestión de usuarios

Tras crear el proyecto de Ionic en el anterior sprint, en este se ha propuesto el objetivo de implementar al completo las tres primeras interfaces diseñadas: Inicio de sesión, Registro y recuperación de contraseña.

Como primer paso, se ha procedido a generar desde la propia terminal de Visual Studio Code cada una de las páginas nuevas que se van a necesitar.


```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS  1
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS F:\Documents\GitHub\LocationNoticer\location-noticer> ionic generate page recuperar-contrasena
```

Figura 38. Comando para generar una nueva página desde terminal en Ionic
(Fuente propia)

Y, como serán necesarios desde este momento en adelante, se crean también los modelos de Usuario y Protegido tal como están formados en la base de datos Firestore Database.

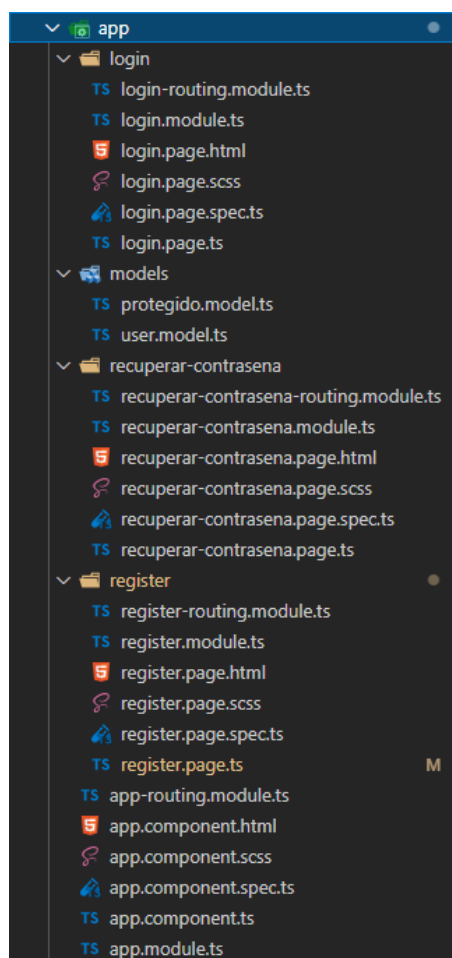


Figura 39. Vista de las nuevas páginas en el directorio
(Fuente propia)

A partir de este punto prácticamente todo el tiempo restante del sprint ha sido dedicado a implementar el código de la aplicación, visualizarlo y comprobar su correcto funcionamiento.

Empezando por configurar los colores corporativos como colores “primary” y “secondary” de la aplicación en el documento `global.scss`, escrito en Sass, un lenguaje de programación para hoja de estilos en cascada (CSS) que utiliza Ionic en sus proyectos. También desde ese archivo se ha cargado la familia de la fuente KoHo que se decidió utilizar en la aplicación en el diseño.

La primera página implementada ha sido la de registro, desde la cual se debe recoger la información del nombre, email, contraseña y tipo de usuario del usuario a almacenar. La aplicación coge esta información y comprueba que sea correcta en sus formatos. Además, tras hacer la conexión con Firebase, esta información será almacenada en la Authentication habilitada para email y contraseña utilizando la función `createUserWithEmailAndPassword()` de `AngularFireAuth` y también en la base de datos `Firestore Database` con funciones de `AngularFirestore`, ambas incluidas en la librería `AngularFire` mencionada en el anterior sprint.

También, al finalizar ese registro, se envía un email de confirmación desde Firebase con la función `sendEmailVerification()` para aumentar la seguridad de la aplicación. El email se genera y envía automáticamente desde Firebase, aunque su plantilla si puede ser modificada. En el email se puede acceder a un enlace generado por Firebase, pero externo al diseño de la propia aplicación desde el cual se confirma automáticamente la cuenta y queda así guardado.

```
TS register.page.ts 1 M X
src > app > register > TS register.page.ts > RegisterPage > ngOnInit

31 async register(user: User){
32   if(this.formValidation()){
33     //Show loader
34     let loader = this.loadingCtrl.create({
35       message: "Please wait..."
36     });
37     (await loader).present();
38
39     try{
40       var dataId;
41       await this.firestore.collection("Usuario").add(user) //Nuevo usuario en base de datos
42       .then(data => {
43         console.log(data);
44         dataId = data.id;
45       });
46       if(user.tipo == 1){ //Nuevo usuario en la subcolección de Protegidos en base de datos
47         await this.firestore.collection("Usuario").doc(dataId).collection("Protegido").add(this.protegido)
48         .then(data => {
49           console.log(data);
50         });
51       }
52       await this.afAuth.createUserWithEmailAndPassword(user.email, this.password) //Nuevo registro en Firebase Authentication
53       .then(data => {
54         data.user.sendEmailVerification(); //Envío de email de verificación de usuario
55
56         //redirect
57         this.showToast("El registro se ha completado correctamente. Revisa tu correo electrónico para confirmar la cuenta.");
58         this.navCtrl.navigateRoot("login");
59       });
60     } catch(e){
61       this.showToast(e);
62     }
63
64     //dismiss loader
65     (await loader).dismiss();
66   }
67 }
68
```

Figura 40. Guardado del usuario del registro en Firebase Authentication y Firestore Database (Fuente propia)

Se ha comprobado el funcionamiento y es correcto en ambas funcionalidades de Firebase, por lo que se ha procedido a generar la página de inicio de sesión o login, que será la página principal o home nada más abrir la aplicación.

En esta interfaz se comprueba que el email y contraseña introducidos sean correctos y existan en Firebase Authentication, para entonces iniciar sesión con la función *signInWithEmailAndPassword()* de AngularFireAuth, en la que se comprueba si la cuenta ha sido verificada vía email.

Desde el inicio de sesión se puede observar un enlace para la posibilidad de que usuarios no recuerden su contraseña, este enlace los llevará a la última página a implementar en este sprint, la página de recuperación de la contraseña.

En esta última página únicamente se ve un mensaje explicando el funcionamiento de la recuperación de contraseña en la app, además de poder introducir la dirección de correo electrónico del usuario que no recuerda su contraseña para poder enviarle un email para poderla cambiar.

Firebase, al igual que ocurre con el email de verificación del registro, también ofrece el servicio de enviar el email para cambiar la contraseña directamente, con la función *sendPasswordResetEmail(email)*.

Cuando este email ha sido enviado, la app automáticamente redirige al usuario a la página inicial, también implementado al finalizar el registro, dejándole un mensaje de que la acción ha sido realizada correctamente, aunque también mostrando mensaje en caso contrario.

Finalmente, se ha implementado el estilo de cada una de las interfaces, tratando de copiar su diseño a máximo posible como se puede observar en la siguiente figura con distintas capturas del resultado, en distintos tipos de pantallas de diferentes dispositivos como ofrece la opción de Herramientas para desarrolladores de Google Chrome.

The figure displays three mobile application screens side-by-side:

- Location noticer:** Features a logo with 'Ln' and a location pin icon. It includes input fields for 'Email' and 'Contraseña' (Password), a link for 'He olvidado la contraseña' (I forgot the password), a blue 'Iniciar sesión' (Log in) button, a question '¿Aún no te has registrado?' (Haven't you registered yet?), and a blue 'Registrarse' (Sign up) button.
- Registro:** Contains input fields for 'Nombre*' (Name), 'Email*', 'Contraseña*' (Password), and 'Repetir contraseña*' (Repeat password). Below these is a toggle switch between 'Guardián' (Guardian) and 'Protegido' (Protected), a note 'Todos los campos que contengan * son obligatorios.' (All fields with * are mandatory), and a blue 'Registrarse' button.
- Recuperar contraseña:** Includes an instruction: 'Introduce tu dirección de correo electrónico, a esa dirección se enviará un enlace desde el que podrás introducir una nueva contraseña.' (Enter your email address, a link will be sent to that address from which you can enter a new password). It has an 'Email' input field and a blue 'Enviar email' button.

Figura 41. Interfaces 1, 2 y 3 implementadas
(Fuente propia)

9.3. Sprint 3: Enlace entre dispositivos a través del código

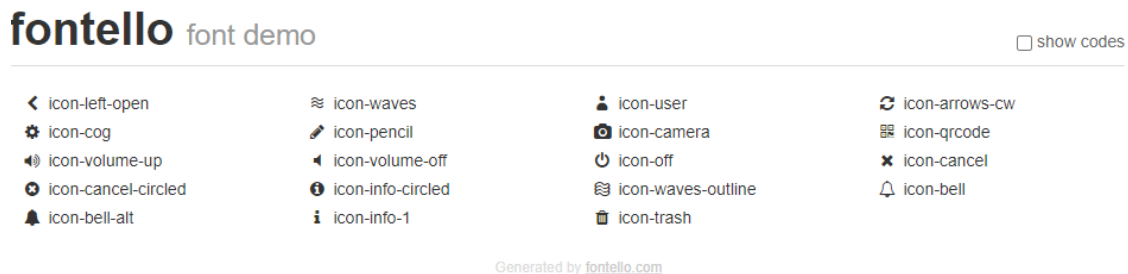
En este sprint se ha marcado como objetivo lograr finalizar las interfaces necesarias para el enlazamiento de dos dispositivos. Serían las interfaces 4 y 5, la interfaz de introducción del código para un Guardián y la interfaz de seguimiento del usuario Protegido respectivamente, además de todas las acciones necesarias desde esas interfaces hacia la base de datos.

Antes de generar las nuevas páginas, se ha tratado de utilizar iconos procedentes de la web de Font Awesome [47], los cuales dieron problemas en su dibujado y, por motivos de versiones, no ha sido posible incluirlos en el proyecto. Tras la búsqueda de alternativas, se decidió descargar físicamente iconos de Fontello [48] e incluirlos desde las hojas de estilo CSS en el proyecto, siendo el resultado de esto el siguiente en las tres primeras interfaces implementadas.



Figura 42. Ejemplo del uso de iconos de Fontello
(Fuente propia)

Además de los iconos visibles en el ejemplo anterior, se han descargado una cantidad mayor con el objetivo de tener desde este momento ya incluidos en el proyecto todos los que puedan ser necesarios. En la siguiente figura se puede observar todos los iconos elegidos.



*Figura 43. Iconos descargados e incluidos en el proyecto
(Fuente: <https://fontello.com/>)*

Ya finalizada la inclusión de estos iconos, ya se ha procedido a la creación de implementación de las dos nuevas interfaces anteriormente mencionadas. Ambas, denominadas en su creación como generar-codigo e introducir-codigo, están destinadas para usuarios Protegido y Guardián respectivamente, por lo que se deberá controlar qué tipo de usuario ha iniciado sesión desde login para poder redirigirlo a la página que le corresponda.

Desde la librería de AngularFireAuth es posible consultar si la sesión está iniciada y por qué usuario con la variable `currentUser` y con `currentUser.email` su dirección de correo electrónico. Además, consultando su tipo se redirige a otra página con la función `navigateRoot()` de la librería NavController [49] del propio Ionic, incluyendo entre paréntesis el nombre de la página a la que se pretenda dirigirse.

También se han implementado todos los mensajes modales de información en las páginas que fueran necesarios, para ser mostrados al pulsar sobre el icono de información. Y se ha implementado una navegación hacia la última página visitada con el icono de la flecha, utilizando la función `back()` de la librería ya mencionada NavController.

Figura 44. Ejemplo de mensaje modal en Location Noticer
(Fuente propia)

Ya de vuelta a la implementación de las páginas nuevas, concretamente empezando por la página generar-codigo, desde ella se generará un código aleatorio de 6 dígitos entre letras mayúsculas, minúsculas y números. Al generar el código también se crea un documento en la tabla de Seguimiento de la base de datos con toda la información por defecto excepto el usuario Guardián, incluyendo el código generado como parte del identificador del documento.

También está incluida una opción de refrescar y generar un nuevo código, lo que implica que el documento en la base de datos queda borrado y se genera uno nuevo con las nuevas características.

Además, en la sección inferior de esta interfaz se puede visualizar una lista de todos los usuarios Guardián que están enlazados realizando seguimiento sobre ese usuario, mostrando sus nombres y un botón para poder romper el enlace de seguimiento y, de esta forma, eliminar el documento de esta relación de la colección de Seguimiento de la base de datos.

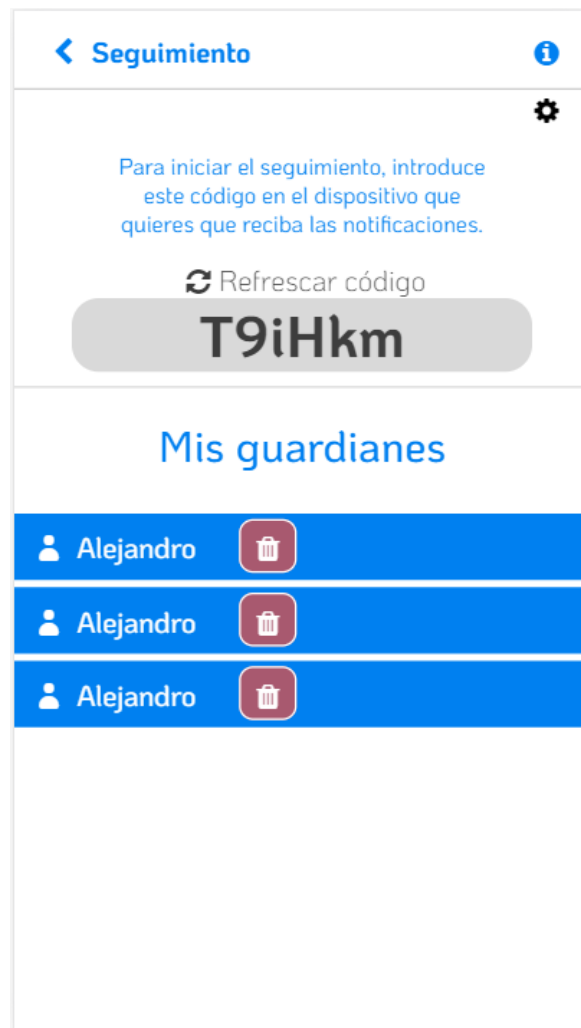


Figura 45. Visualización de la página generar-codigo
(Fuente propia)

Ambas acciones se realizan con la librería de AngularFireStore, utilizando las funciones `collection()` pasándole el nombre de la colección por parámetro como una cadena de texto para recibir todos los documentos de la colección y, aplicando a esta las funciones `doc()` con el identificador del documento pasado por parámetro y `delete()` a la que te devuelve el documento, se elimina por completo el documento indicado de esa colección. En la siguiente figura se puede observar el ejemplo de uso de estas funciones en el código implementado de la aplicación.

```

this.firestore.collection("Seguimiento").snapshotChanges().subscribe(data => {
  let docs = data.map(e => {
    if(miid == e.payload.doc.data()["protegido"]){
      return [e.payload.doc.data()["guardian"], e.payload.doc.id];
    }
  });
});

```

```

borrarGuardian(id){
  try{
    this.firestore.collection('Seguimiento').doc(id).delete().then(()=>{
      this.showToast('Se ha eliminado el enlace correctamente.');
```

*Figura 46. Consulta y borrado de colecciones y documentos en Firestore Database
(Fuente propia)*

La página del usuario Protegido generar-código es la interfaz general o principal cuando este ha iniciado sesión, no es necesario redirigir a ninguna otra página y desde ella se podrán realizar el resto de acciones que fuesen necesarias para este usuario, como el acceso a la página de configuración que será desarrollada más adelante pero cuyo icono ya está incluido.

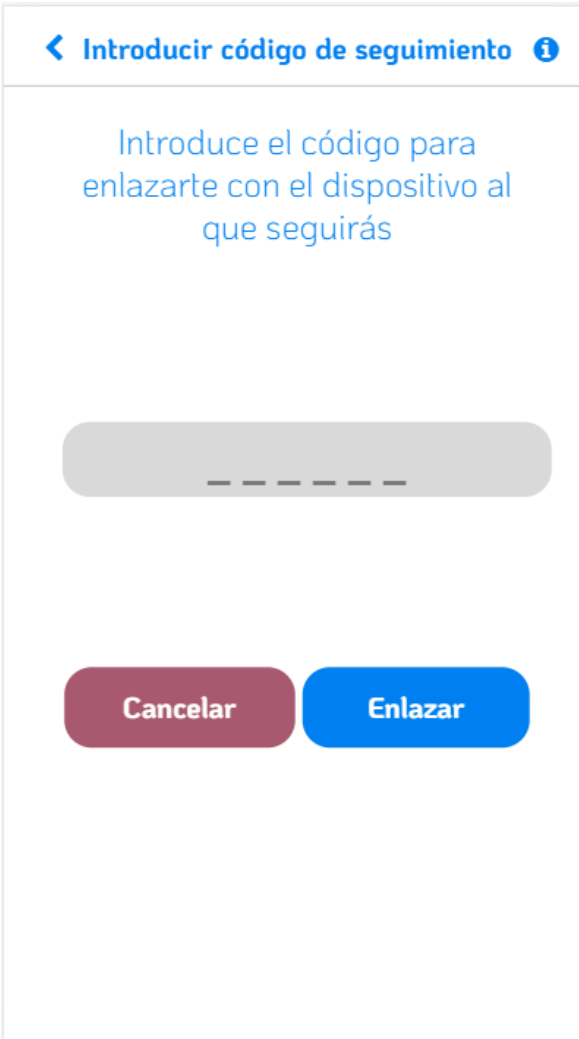
Ya finalizada la interfaz de generar-código, se procede a empezar la implementación de la interfaz introducir-código. En ella únicamente se puede introducir el código de 6 dígitos, en caso de no tener ese número concreto de dígitos se indicará al usuario por pantalla.

Tras pensar detenidamente como funcionaría esa finalización del enlazado en la base de datos, se decidió hacer una consulta a la base de datos solicitando el documento que coincida con el código introducido y, posteriormente, editar la información de dicho documento incluyendo el usuario Guardián que no fue incluido en su creación desde la página generar-código. Todo esto sería ejecutado al pulsar el botón de Enlazar incluido en la interfaz.

Tras la implementación de la lógica definida en el párrafo anterior, surgió un error imprevisto, todo el código que estaba incluido dentro de la función que contenía ambas peticiones seguidas, la de obtención del documento y la de edición, se ejecutaba en bucle de forma infinita sin forma alguna de cortar esta ejecución. Tampoco había otra forma de desarrollar en código esta función debido a la asincronía necesaria para estas funciones y así obtener la información necesaria que se recibe desde una función para poder ser utilizada en la siguiente.

Finalmente, se ha tomado la decisión de separar esas dos acciones en dos funciones distintas y separadas y que sean ejecutadas desde puntos distintos. En primer lugar, la lectura del código y la solicitud de su documento a base de datos se realiza con el evento que recibe la aplicación al levantar una tecla del teclado pulsada, haciendo la petición a base de datos únicamente cuando recibe este evento y el código recibido es de 6 caracteres y así se almacena primero el documento.

Y, por último, tras pulsar el botón Enlazar, únicamente se edita tal documento en base de datos, utilizando la función *update()* de AngularFireStore, pasando por parámetro la información a editar o incluir sobre el documento ya creado.



Introducir código de seguimiento ⓘ

Introduce el código para enlazarte con el dispositivo al que seguirás

Cancelar Enlazar

*Figura 47. Visualización de la página introducir-codigo
(Fuente propia)*

Tras comprobar el correcto funcionamiento de esta última página tras arreglar el error producido inesperadamente, se puede ver cómo quedaría en su versión final la página visualmente en la figura anterior.

Después del correcto enlazado de seguimiento, el usuario Guardián será redirigido a la página que incluya el mapa para poder seleccionar el rango, esta página será implementada en sprints posteriores.

9.4. Sprint 4: Inclusión del mapa

Tras finalizar la implementación de las interfaces del sprint anterior, en este sprint se ha apuntado como objetivo implementar una base para la nueva página denominada mapa y la selección/edición del rango en esta.

Como se puede observar en el apartado de diseño, la página mapa tiene tres estados: uno general de seguimiento, otro para la selección o edición del rango de seguridad y otra prácticamente igual a la primera pero que marca la posición real del usuario Protegido, ya que esta última se podrá visualizar únicamente si ha saltado la alarma.

En primer lugar, se estudió la documentación de Mapbox [50] en su web, para conocer cómo insertar el mapa en la aplicación y sus rangos o marcadores en el mapa. Tras observarlo detenidamente y estudiar otras alternativas que también usaban Mapbox, se decide finalmente por hacer uso de Leaflet [51], una librería que hace uso de los mapas de Mapbox y que, además, permite la implementación de sus mapas con mayor facilidad y con un código más simple de programar.

Con el fin de tener el proceso de registro de un usuario Guardián finalizado, se implementó las funcionalidades de la página en su estado de selección o edición del rango. Esta interfaz contiene dos botones, para cancelar la selección y mantenerse en el estado anterior, aunque no tuviese ningún rango guardado, y para confirmar el rango seleccionado y guardarlo.

Además, se ha introducido un botón para refrescar la página del mapa. Un botón incluido únicamente de forma temporal para la etapa de desarrollo.

El funcionamiento de la página tiene una lógica simple. Cuando el mapa recibe el evento de que se ha hecho click sobre este, se recoge de este evento las coordenadas geográficas de longitud y latitud, las cuales son utilizadas como centro del rango (o, como es denominado por Leaflet en

su documentación, círculo o circle) y el radio de este será el rango seleccionado en el input de tipo slider⁶, actualizándose el rango seleccionado de manera automática mientras se puedan editar ambas selecciones.

```
//Inicializar mapa de Leaflet
this.mapaLeaflet = L.map('mapa').setView([this.centerMapCoordinates[1], this.centerMapCoordinates[0]], 18);
L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token={accessToken}', {
  attribution: '© <a href="https://www.mapbox.com/">Mapbox</a>',
  maxZoom: 22,
  id: 'mapbox/streets-v11',
  tileSize: 512,
  zoomOffset: -1,
  accessToken: environment.mapboxKey
}).addTo(this.mapaLeaflet);

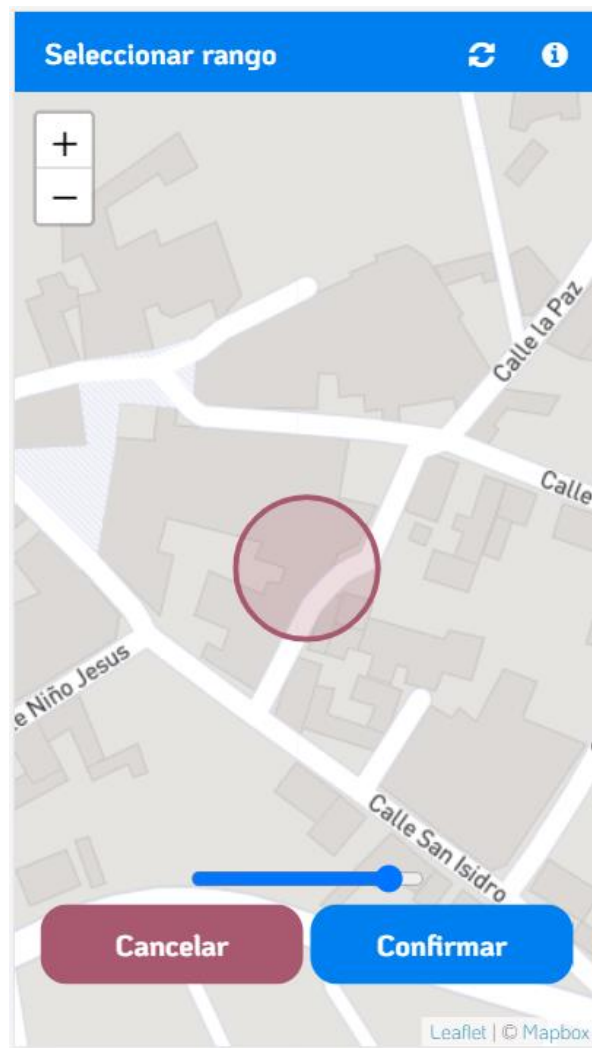
this.circleRango = L.circle([this.circleCoordinates[1], this.circleCoordinates[0]], {
  radius: this.distanciaRango,
  color: '#a8596f'
}).addTo(this.mapaLeaflet);
```

*Figura 48. Inicialización del mapa con Leaflet y creación de un circle
(Fuente propia)*

Además, en el caso de tener ya información almacenada sobre el rango, esta se reproducirá por defecto al entrar a la página, ya sea para seleccionar o editar el rango como para el resto de los estados de la interfaz, centrando de esta forma el mapa en el rango y no en la posición real del usuario.

Cuando se confirma un rango, este es subido a la base de datos al documento correspondiente de la colección de Seguimiento y, automáticamente, el estado de la interfaz pasa a ser de seguimiento, siendo ya imposible editar desde este estado el rango.

⁶ Slider: En castellano, deslizador.



*Figura 49. Visualización de la página mapa con estado de selección de rango
(Fuente propia)*

Dado el avance, ya obteniendo el mapa desde la aplicación e interactuando con él, el objetivo marcado para los siguientes sprints, ya encarrilando los finales, se centra en: recoger la ubicación real del usuario desde la aplicación, generar una APK⁷ para poder hacer comprobaciones más reales desde dispositivos físicos y finalizar el resto de las interfaces restantes del diseño que aún no han sido implementadas.

⁷ APK (Android Application Package): extensión de un archivo para instalar aplicaciones en el sistema Android.

9.5. Sprint 5: Seguimiento con ubicación real

En este sprint se ha decidido por lograr obtener la ubicación real de un dispositivo en tiempo real, para así lograr implementar la lógica de la funcionalidad principal del proyecto.

Esto, se ha logrado con el plugin propio de Ionic, Geolocation [52], desde el cual se obtiene la posición en coordenadas de latitud y longitud simplemente con la función *getCurrentPosition()*.

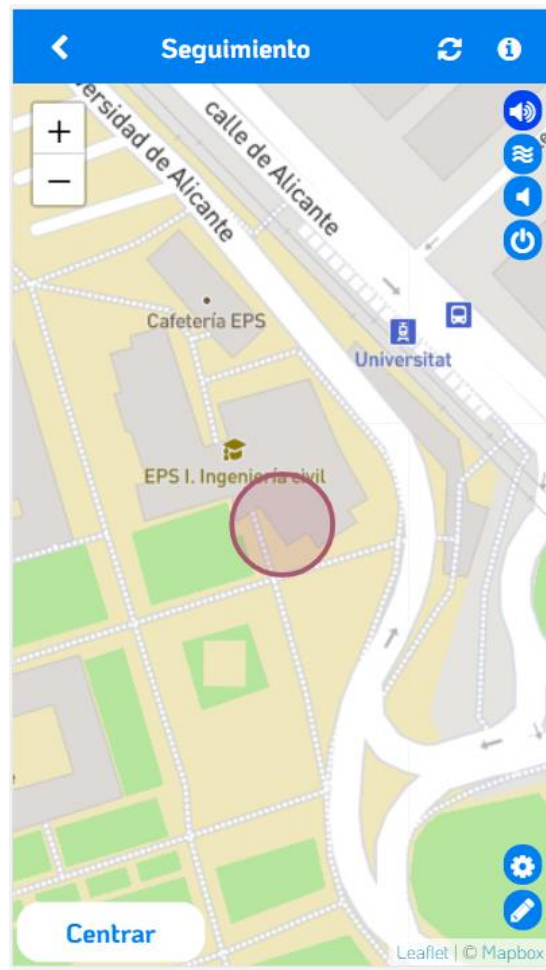
Esta función ha sido incluida como parte de las páginas mapa (para su estado de Seguimiento o Alarma) y generar-codigo.

En la primera, su función es centrar el mapa nada más se inicialice siempre y cuando no haya un rango seleccionado, en caso de haberlo se centrará en este. También, si no logra recopilarse dicha información antes de iniciarse, el mapa se centrará de una forma general mostrando el mapamundi prácticamente completo.

En la segunda, se recoge la ubicación del usuario Protegido almacenándola y actualizándola constantemente de manera periódica en la base de datos.

Aprovechando la implementación de parte de la lógica de la interfaz mapa en su estado de seguimiento, también se ha implementado su diseño con pequeños cambios respecto al diseño final con Figma en el apartado de Diseño de interfaces.

Ha sido añadido un botón Centrar, el cual devuelve el mapa, con la función *flyTo()* de Leaflet, a la posición geográfica en la que se encuentra el rango. Además, debido a que el mapa de Leaflet incluye automáticamente dos botones en la parte superior izquierda de la pantalla para ampliar o disminuir el zoom sobre el mapa, los botones asociados al estado de sonido y activación de la alarma se sitúan ahora en la zona superior derecha de la interfaz y el acceso a la edición de rango y a la página de configuración se encontrarán en la zona inferior derecha. El resultado final de estos cambios y de la interfaz en dicho estado es el de la siguiente figura.



*Figura 50. Visualización de la página mapa con estado de seguimiento
(Fuente propia)*

Además, se ha creado la página config para la configuración, aunque de manera temporal esta únicamente incluye un botón de cerrar sesión para poder hacer futuras pruebas tanto en navegadores web tanto en dispositivos móviles reales durante la implementación.

Para terminar con la lógica del estado de seguimiento de la interfaz mapa, se compara la información de la geoposición central y distancia radial de la zona segura o rango con la información de la posición en tiempo real almacenada en la base de datos, concretamente en la subcolección Protegido de la colección Usuario. Si la distancia entre ambos puntos, el centro del rango y la posición real del usuario Protegido, es mayor que la distancia radial, significará que está fuera del rango y por lo tanto tendrá que saltar la alarma. Aunque de manera temporal, simplemente salta un mensaje por pantalla indicándolo.

Además, tras implementar todo lo mencionado anteriormente y tal como se indica al final del apartado referente al Sprint 4, se toma la decisión en este momento de generar la APK. Para así

comprobar el diseño y funcionamiento de la aplicación en dispositivos físicos que serán los destinatarios para el uso de ésta.

Para generar la APK del proyecto de Ionic, haciendo uso de Cordova, se utiliza el comando *ionic cordova build android*. A raíz de este simple comando han surgido una gran cantidad de errores y problemas del proyecto e incluso de la propia configuración del ordenador desde el que se está implementando y desarrollando por completo este proyecto.

La operación Build ejecutada automáticamente dentro del comando saltó con un error, no encontraba en el sistema el JDK⁸ [53] y el SDK (en este caso más conocido como el programa Android Studio [54]) de Android, por lo que ambos fueron instalados.

Tras su instalación tampoco los reconocía, por lo que, para subsanar distintos errores encadenados, se incluyeron varias rutas a las carpetas destino de la instalación de ambos programas como variables de sistema en Windows. Se crearon las variables ANDROID_SDK_ROOT, ANDROID_HOME, _JAVA_OPTIONS y todas ellas también se incluyeron dentro de la variable Path.

Tras lograr completar correctamente la instalación y configuración de variables de sistema mencionadas, saltó un error referente a que no encontraba Gradle [55], un software que automatiza los build (o construcciones de aplicaciones) y que suele instalarse con Android Studio de forma conjunta. Tras comprobaciones, se confirmó que no se había realizado dicha instalación y se procedió a instalarlo por separado. Aún de esa manera, la ejecución del build siguió dando problemas, aunque estos se solucionaron tras borrar carpetas autogeneradas por la instalación pero que torpedeaban la lectura de otras y modificar las propiedades de la configuración de Gradle dentro del proyecto.

Además, siguió fallando la ejecución, a lo cual se procedió a acceder a la herramienta SDK Manager incluida en Android Studio para instalar Android SDK Command-Line Tools, en su última versión. Tras esta última instalación, finalmente se generó un archivo APK con el proyecto.

⁸ JDK (Java Development Kit): Software que proporciona herramientas para el desarrollo de programas en Java.

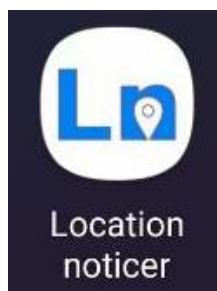


Figura 51. Captura de la aplicación desde el menú de un teléfono Android
(Fuente propia)

Se ha comprobado el correcto funcionamiento de las funcionalidades y páginas ya implementadas. El resultado de estas pruebas de forma general ha sido correcto y satisfactorio y a partir de ese momento se probará los avances en la implementación desde el navegador web a la par que se escribe el código y en un dispositivo real Android cuando funcione correctamente en el anterior.

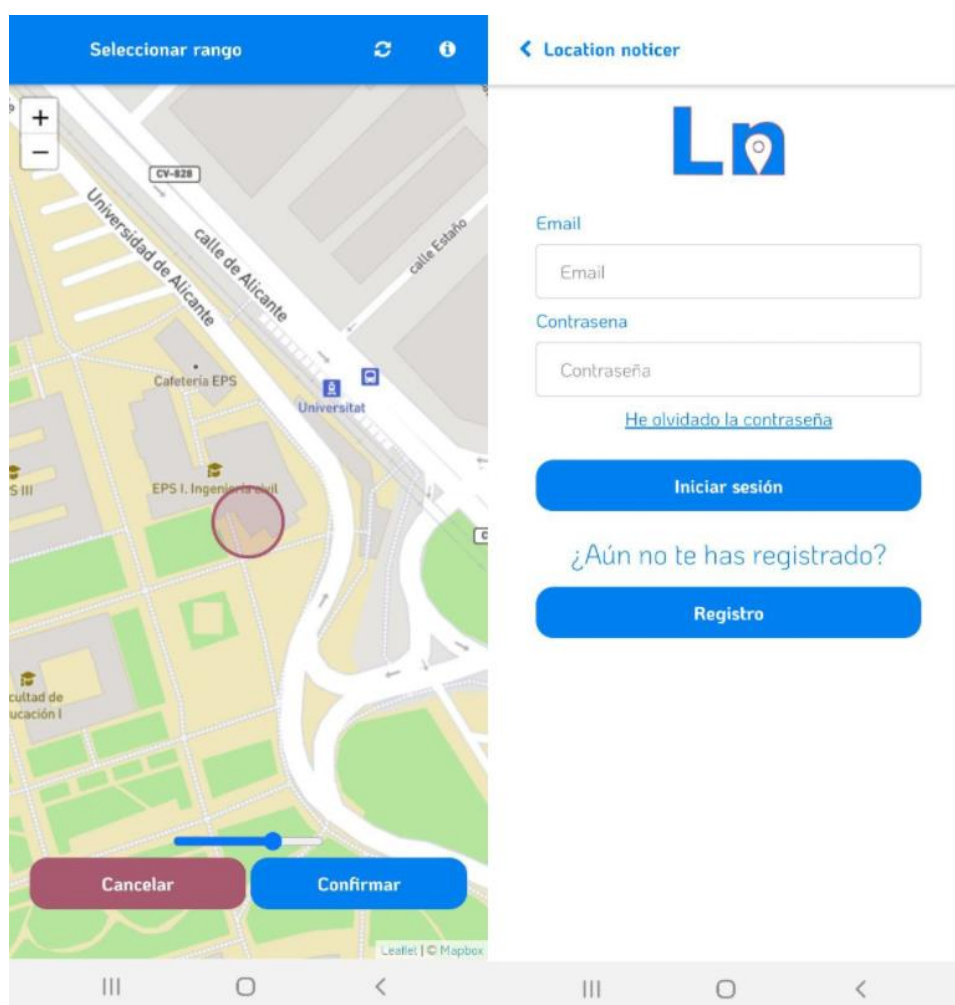


Figura 52. Capturas de las interfaces de la APK vistas desde un Samsung Galaxy A50
(Fuente propia)

9.6. Sprint 6: Alarma y configuración

En este sprint, el desarrollo de la aplicación se ha centrado en la alarma, tanto como estado de la página mapa como página en sí, y en la página de configuración.

Como se menciona en el sprint anterior, desde la página mapa se hacía una comprobación entre las posiciones del rango, su distancia y la posición que se recibe del usuario Protegido, y cuando se daba la situación que debía saltar la alarma se mostrar un mensaje por pantalla de forma temporal hasta que se diesen más avances en la implementación.

En este sprint se ha implementado la nueva página Alarma, a la cual se accederá nada más salte la alarma, siempre que no estuviese activada esta ya anteriormente.

La página es simple, consta de un título en el que, dependiendo de si el usuario tiene o nombre almacenado, muestra el nombre del usuario Protegido, un icono que hace referencia a la alarma y, en lugar de los botones que se incluían en el diseño de interfaces, se ha incluido un texto que indica la forma de acceder a la visualización de la última posición en vivo del usuario Protegido.

Este cambio respecto al diseño se debe a que, de los dos botones diseñados, uno todavía no está implementado, y según el tiempo restante de implementación del proyecto podría decidirse como futura mejora, y el otro haría la misma acción que hace el botón para salir de la página, que sería volver a la interfaz mapa, pero con estado de alarma.

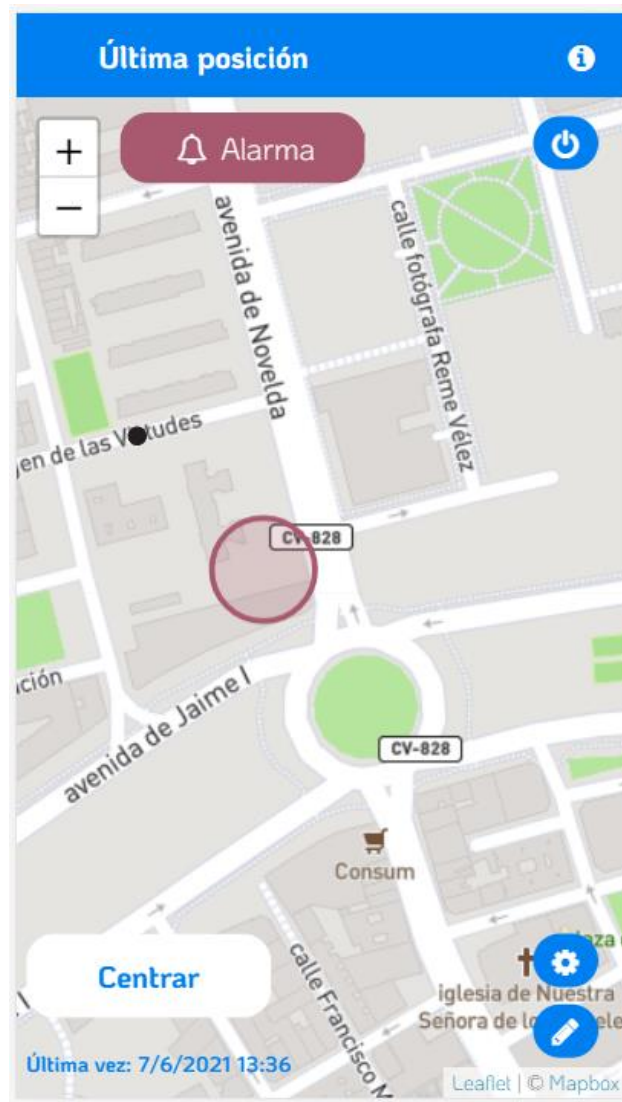


*Figura 53. Visualización de la página de alarma
(Fuente propia)*

Tras finalizar la implementación de la página de la alarma, se ha procedido a implementar el estado de alarma en la interfaz mapa.

Esta parte como base del estado de seguimiento de la interfaz, a la que con los mismos botones se le incluye un aviso visual de que el seguimiento se encuentra en estado de alarma, el botón central ya no centra el mapa hacia el rango, si no hacia la posición actual del usuario Protegido, la cual está dibujada en el mapa por un punto.

Esto se debe a que cuando no salta la alarma, esta se centra en el dibujo del rango, ya que el usuario seguido se encuentra en el interior, aunque no se dibuje. Además, también se indica por escrito en la parte inferior de la interfaz cuándo fue la última vez que se recibió la posición del Protegido.



*Figura 54. Visualización de la página mapa en estado de alarma
(Fuente propia)*

Como se puede observar, también se pueden visualizar en la anterior figura cambios respecto a los botones del estado de la alarma. Debido a la decisión de implementar la opción de cuatro estados (con sonido, en modo vibración, silencio y apagada) en un futuro, se ha optado por eliminar todos los relativos al sonido o volumen y únicamente dejar un botón desde el cual poder conectar y desconectar la alarma.

Con la implementación de la alarma completada, en su propia interfaz y en la del mapa, el siguiente paso en la implementación era desarrollar la única interfaz restante, la interfaz de la página config.

Esta interfaz se creó en el sprint anterior para realizar pruebas pudiendo cerrar sesión. En este sprint, se ha implementado el resto de la página.



*Figura 55. Visualización de la interfaz config para un usuario Protegido y para un Guardián
(Fuente propia)*

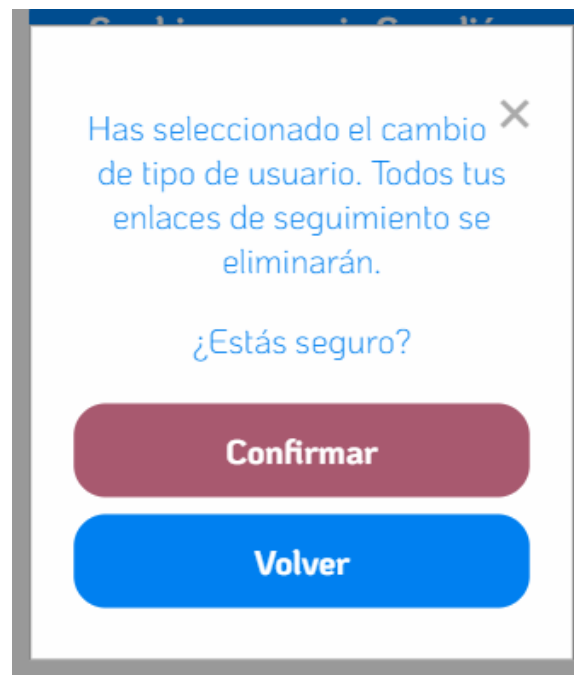
En ella se puede observar que no incluye todo lo que se incluyó en su diseño. Esto es debido a que la parte de edición de la cuenta del usuario, en la que se incluía edición de nombre, email y contraseña, no ha sido incluida en este momento, aunque sí se hará en el futuro.

Tampoco ha sido incluida la opción de eliminar cuenta, debido a que la librería de Firebase no contiene la única función de toda la documentación de Firebase que permite la eliminación de usuarios de Firebase Authentication. Por ese motivo, lamentablemente no ha sido posible su implementación, aunque en versiones posteriores de esta librería se tratará de implementar en caso de sí ser incluida.

Tampoco aparecen las funcionalidades de silenciar alarma y editar rango, puesto que se puede acceder a ellas desde la interfaz mapa y se podría ahorrar el paso intermedio de acceder a la página de configuración para ello.

Con todos esos cambios, el resultado es el visualizado en la figura anterior, donde la parte superior de la interfaz podrán visualizarla ambos tipos de usuarios y la parte inferior estará disponible únicamente para usuarios de tipo Guardián.

Por último, para todos los botones de la configuración, excepto para el cierre de sesión, se deberá confirmar que se quiere realizar dicha acción contestando a un mensaje modal, para así evitar posibles errores e informar de que ocurrirá si se ejecutase.



*Figura 56. Visualización de un mensaje modal de confirmación en la página config
(Fuente propia)*

9.7. Sprint 7: Últimos detalles y corrección de errores

Llegado este sprint, todas las interfaces han sido implementadas, aunque no todas las funcionalidades de estas, puesto que algunas serán implementadas como trabajo futuro y se hablará de ellas en su respectivo apartado de la memoria.

El objetivo de este último sprint de la implementación es finalizar funcionalidades no implementadas hasta este momento, retocar pequeños cambios en interfaces, siempre que sean necesarios, y subsanar errores que se han ido descubriendo durante la implementación de la aplicación.

En primer lugar, se ha implementado la lectura y el almacenamiento del estado de la alarma en la base de datos, ya preparada para ello con la variable activo en la colección de Seguimiento.

Esta funcionalidad, incluida como parte del funcionamiento de la interfaz mapa, consultar nada más abrir la página si la alarma por defecto o en su último guardado se encontraba activada o desactivada, para mantenerla en ese estado.

<div> <div> </div> <div> <div>></div> <div>Seguimiento</div> <div>></div> <div>2c7ik1-1625669...</div> </div> </div>		
<div> <div>locationnoticer</div> <div> <div>+ Iniciar colección</div> </div> </div>	<div> <div>Seguimiento</div> <div> <div>+ Agregar documento</div> </div> </div>	<div> <div>2c7ik1-1625669192393</div> <div> <div>+ Iniciar colección</div> </div> </div>
<div> <div>Seguimiento</div> <div>></div> </div>	<div> <div>2c7ik1-1625669192393</div> <div>></div> </div>	<div> <div>+ Agregar campo</div> </div>
<div> <div>Usuario</div> </div>	<div> <div>59WzEd-1625236527077</div> <div>BSI35Q-1625236527077</div> <div>IJXvmr-1625346453189</div> <div>zPghNQ-1625657468610</div> </div>	<div> <div>activo: true</div> <div>alarma: 0</div> <div>codigo: "2c7ik1"</div> <div>protegido: "angellopezcarrion@hotmail.com"</div> </div>

Figura 57. Documento de la colección Seguimiento tras generar un nuevo código de enlace
(Fuente propia)

Así mismo, cuando el estado cambia, este es editado en la base de datos mediante la función `update()` de AngularFirestore.

En segundo lugar, se termino de implementar la página generar-codigo. En ella, debido que al abrir la página algunas funciones se ejecutaban varias veces, la lista de guardianes del usuario Protegido salía repetida y se podía leer los nombres de cada uno varias veces.

Esto se ha subsanado fácilmente, ya que únicamente se guardan guardianes dentro de la lista que se dibuja en el código HTML si estos no estaban ya dentro. Y si algún guardan es eliminado, esta lista se regenera, dejando fuera de ella obviamente al usuario eliminado.

Además, también se ha incluido un modal como las usados en la página config para confirmar la eliminación de un guardián, de ese modo evitando que un solo click elimine un seguimiento que podría ser importante.

También resultaba un problema la generación de nuevos códigos para la base de datos, que se encontraba repleta de documentos vacíos que, puesto que cada vez que se accede a la página se genera uno nuevo, no iban a ser útiles ya.

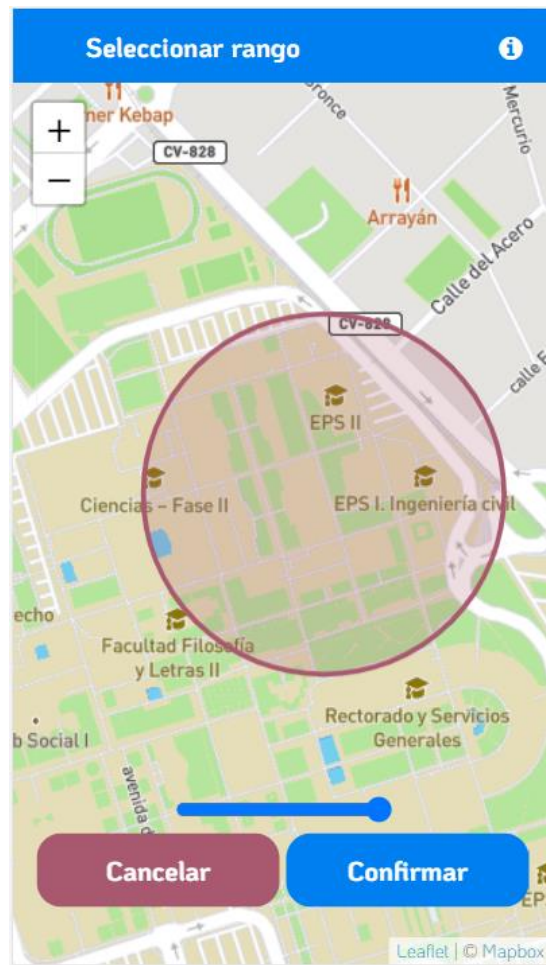
Para solucionar este problema, cuando se genere un código de seguimiento nuevo y, a su vez, un documento nuevo en su colección de la base de datos, se eliminarán todos los documentos con código que finalmente no hayan sido enlazados y que sean del usuario Protegido que crea el código nuevo. De esta forma siempre habrá como máximo un documento sin enlazar por cada usuario Protegido.

Siguiendo con las corrección y pequeños cambios, para dejar la aplicación bien conectada y que no puedan ocurrir problemas en la navegación, se ha revisado que páginas deben o no tener el botón para retroceder de página.

Se ha eliminado la acción de retroceder de las páginas que implican la configuración de un guardián con selección de rango y de aquellas que son principales, no se podrá retroceder de la página login ni la página mapa.

Por último, tras realizar pruebas en dispositivos móviles, se ha comprobado que la ubicación que transmiten estos no es exacta y tiene un margen de error a la hora de obtenerla. Ello conlleva que la ubicación podría verse desplazada lo suficiente como para que salte la alarma sin que en realidad deba hacerlo.

Debido a ese motivo, y teniendo en cuenta como trabajo futuro estudiar la manera de mejorarlo, se ha tomado la decisión de aumentar las distancias mínimas y máximas con las que se podrá seleccionar el rango.



*Figura 58. Ejemplo nueva distancia radial máxima en la selección de rango
(Fuente propia)*

Las nuevas distancias máxima y mínima serán de 200 y 25 metros respectivamente. Por lo que el mínimo podría equivaler a una casa grande o urbanización y el máximo a un conjunto de calles o barrio. Un ejemplo de la nueva distancia máxima se puede visualizar en la figura anterior, en la que se puede ver la distancia aplicada en la Universidad de Alicante.

Finalmente, ya acabada la aplicación por completo, se ha generado de nuevo el archivo APK, que no ha tenido ninguna gran complicación puesto que ya se había generado anteriormente y ya entonces se solucionaron los problemas correspondientes.

10. Pruebas y validación

En este apartado, tras finalizar la etapa de implementación de la aplicación, se llevan a cabo diferentes pruebas y comprobaciones para asegurar su correcto funcionamiento o, en caso contrario, conocer los errores que pudiese haber para poder solucionarlos.

Dado que la aplicación no se encuentra implementada al 100% en su versión final, por la propia dificultad de desarrollar una aplicación de este calibre en un tiempo limitado como es el planteado en un TFG, esta no ha sido publicada en ningún lugar para su libre descarga, como podría ser una tienda de aplicaciones como Google Play Store. Esto implica que el número de personas que la han probado es reducido como para realizar una encuesta o estudio.

De todas formas, sí se han revisado y comprobado qué objetivos y requisitos, ya sean funcionales o no, se han logrado llegado este punto.

En el caso de los requisitos funcionales, o RF, se han cumplido todos. Puesto que se ha logrado habilitar mecanismos para enlazar dispositivos entre usuarios, capacidad para elegir rango y verlo junto al mapa, posibilidad de recibir la alarma cuando esta salte, capacidad de cancelar o desconectar la alarma, posibilidad de consultar la última posición si la alarma hubiese salta y poder gestionar los usuarios. Esta última es posible desde la interfaz ofrecida por Firebase en su web, ya que como ya era un servicio ofrecido por Firebase se decidió no implementarlo en la aplicación.

En cuanto a los requisitos no funcionales, o RNF, se han cumplido todos excepto uno. El requisito RF1 no se ha cumplido en este momento, dado que como este mismo apartado explica, no se ha subido a ninguna plataforma de descarga de aplicaciones debido a no estar finalizada.

Estos requisitos serán revisados durante los avances de la aplicación, puesto que su cumplimiento a estas alturas es temporal y, desde este momento en adelante, en caso de seguir con la implementación, podría cumplirse el requisito incumplido.

Referente a los objetivos personales del proyecto, descritos en el apartado Objetivos de este mismo documento, se hablará y profundizará más en el posterior apartado referente a las conclusiones.

Ya revisados todos los objetivos y requisitos, se procede a hacer las pruebas pertinentes a la aplicación con los servicios que ofrece Firebase.

En primer lugar, haciendo uso de Firebase Crashlytics, una herramienta encargada de realizar informes de fallos en tiempo real, no se detectó ningún tipo de fallo o problema, por lo que el funcionamiento de la aplicación es correcto.

En segundo lugar, haciendo uso de Firebase Performing Monitoring, se recabó información sobre el rendimiento de la app, comprobando el correcto rendimiento de esta. Además, obteniendo mejoras en los tiempos de primer procesamiento de imagen en localhost y tiempo de respuesta de la API de Firestore.

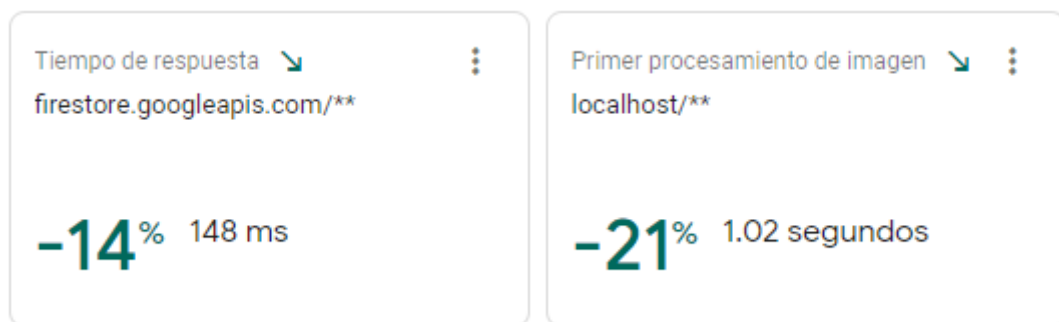


Figura 59. Panel de tiempos de respuesta de Firebase Performance Monitoring
(Fuente propia)

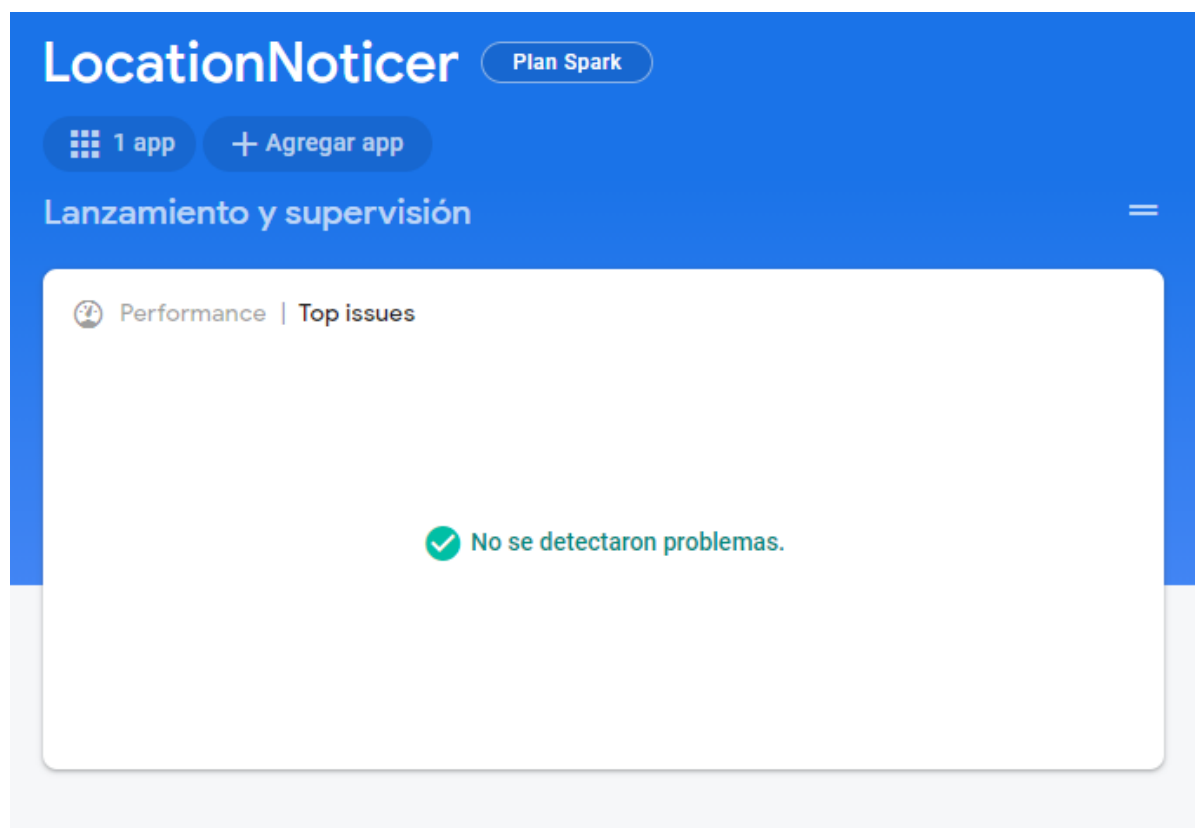


Figura 60. Panel de control de Firebase Performance Monitoring
(Fuente propia)

Por último, en tercer lugar, se ha hecho uso de Firebase Test Lab. Esta herramienta ejecuta la aplicación, la cual solicita que sea cargada como APK en la herramienta, en dispositivos reales o virtuales. Se realizaron 3 pruebas de tipo Robo en 3 dispositivos distintos, concretamente un LG Nexus 5, un Google Pixel 3 y un Samsung Galaxy S9. Todas estas pruebas finalizaron con una correcta ejecución y sin ningún tipo de problema.

Al igual que se ha mencionado en la revisión de requisitos, estas pruebas, aunque positivas, son temporales y será necesario continuar con el testing a la par que se continúe con la implementación de la aplicación, de esta forma más adelante con más información recabada se obtendrán resultados más reales.

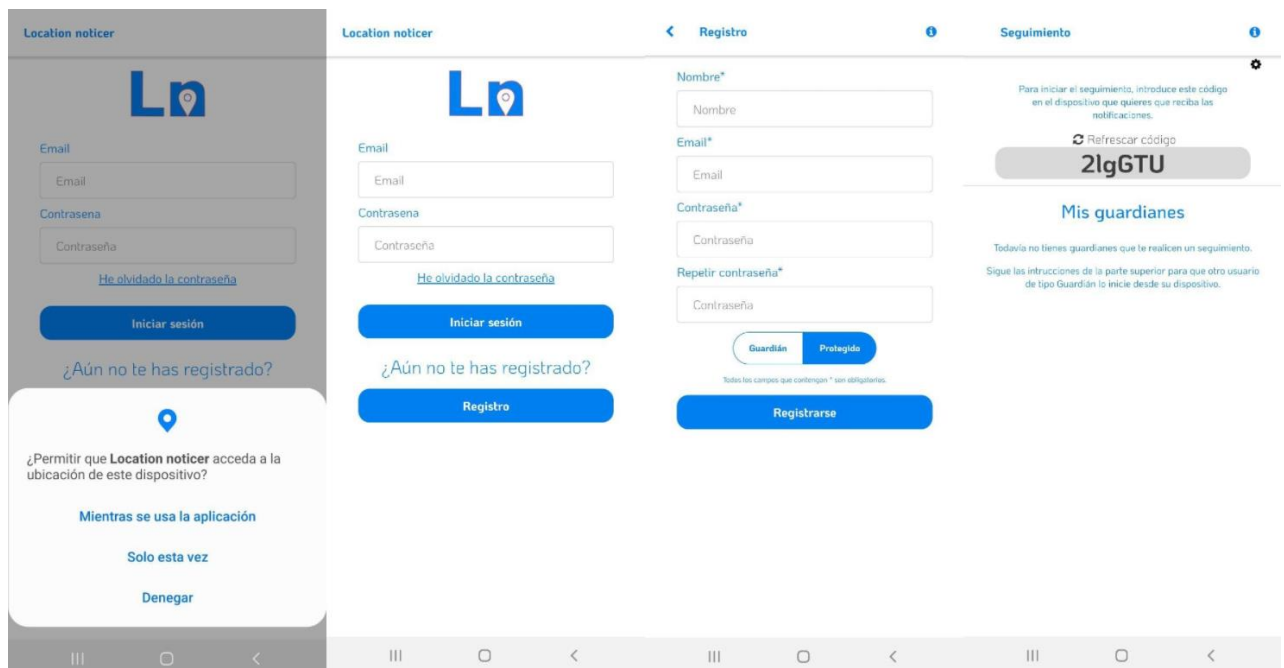
11. Resultados

Este apartado está dedicado a mostrar y analizar los resultados obtenidos en el proyecto, tanto como producto final como el coste temporal que ha conllevado y cuál es la relación con asignaturas cursadas en la carrera.

11.1. Producto final

En este apartado se mostrará, a modo de demostración guiada por imágenes, cuál sería el uso de Location noticer, en su estado actual como mínimo producto viable, en orden de aparición de las interfaces. Además, esta guía podrá servir como base para el video de presentación del proyecto.

En primer lugar, la primera interfaz en mostrarse nada más abrir la aplicación es la de login o inicio de sesión.



*Figura 61. Primeras interfaces para un usuario Protegido de la demostración
(Fuente propia)*

Además, nada más ser abierta por primera vez, se lanza un aviso en el que se solicita acceso a la ubicación del dispositivo para su funcionamiento. Si esta solicitud es rechazada, las

funcionalidades principales de la aplicación no tendrán un correcto funcionamiento, por lo que es indispensable permitir el acceso a la ubicación durante todo el uso de la aplicación.

Al distinguir entre dos tipos de usuarios con funcionalidades o acciones distintas en distintas interfaces, primero se hará el recorrido que debería hacer un usuario Protegido.

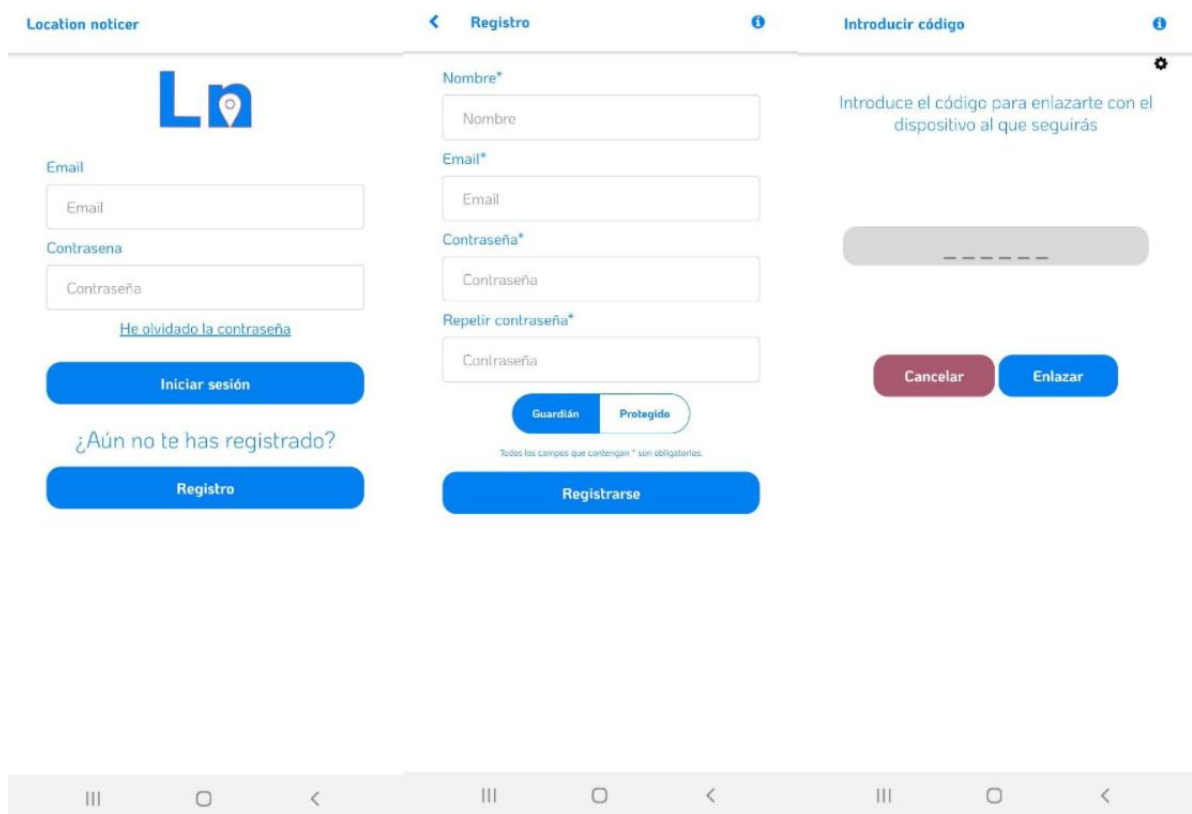
Antes de iniciar sesión, pero si a través de esta página, se accederá a la página de registro en la que se registrará como un usuario Protegido.

Tras iniciar sesión, habiendo tenido que verificar por correo electrónico el registro, se abrirá la interfaz de generación de código y seguimiento de guardianes. Nada más entrar, ya se habrá generado un código para poder enlazar con un usuario Guardián, el cual podrá ser regenerado en cualquier momento.

Además, al encontrarse recién registrado, en lugar de aparecer los Guardianes que hacen el seguimiento, aparece un mensaje de texto. En cuanto haya sido enlazado, se podrás visualizar los usuarios Guardianes que siguen a ese Protegido y poder romper su enlace si fuese necesario.

Para un usuario Protegido, ese es el total de interfaces de las que podrá hacer uso además de la configuración, que no es principal y no se muestra en esta demostración, aunque si en el apartado de implementación.

Sigue la demostración con cómo haría uso de la app un usuario Guardián.



*Figura 62. Primeras interfaces para un usuario Guardián de la demostración
(Fuente propia)*

Al igual que con el usuario Protegido, el Guardián nada más entrar se encontrará la página login y el aviso sobre la ubicación. Este deberá registrarse como Guardián y, al verificar el email e iniciar sesión, se accederá a la interfaz de introducción de código. Esta interfaz solo será visible al iniciar sesión cuando el usuario Guardián no tenga ningún enlace con un usuario Protegido.

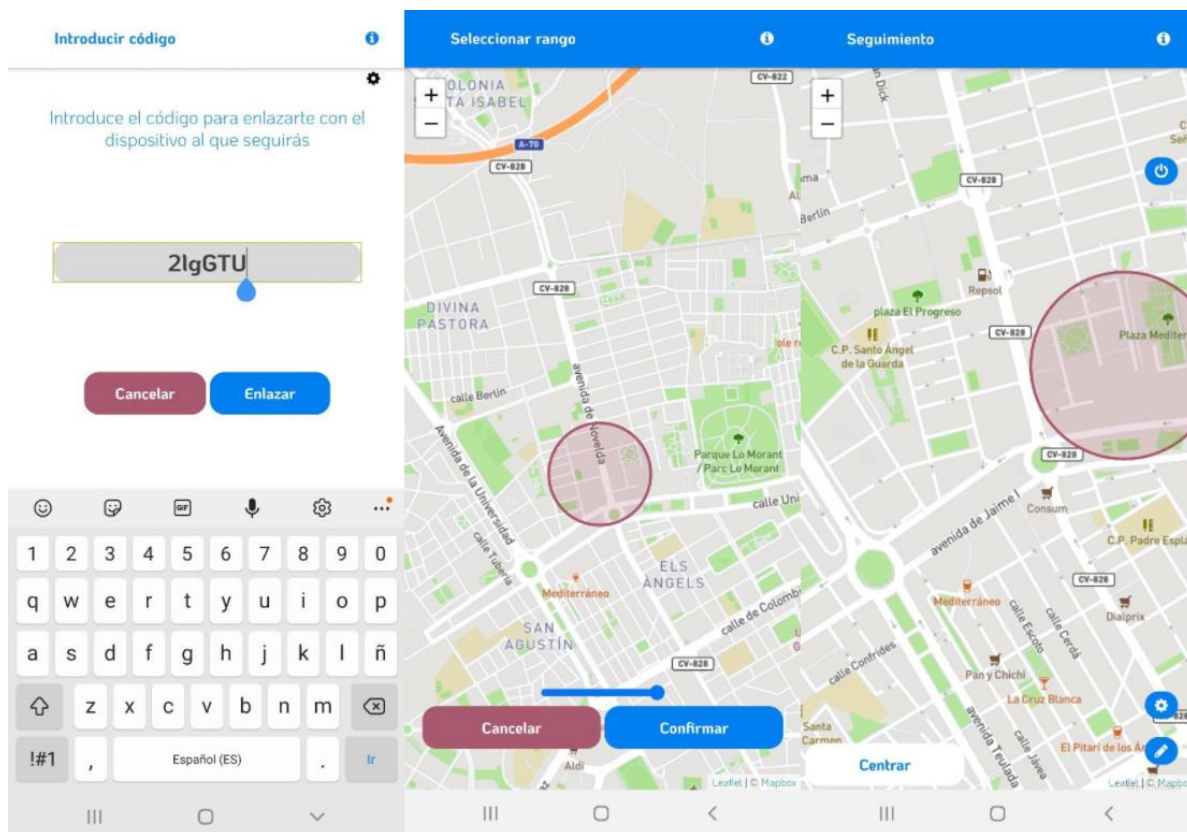
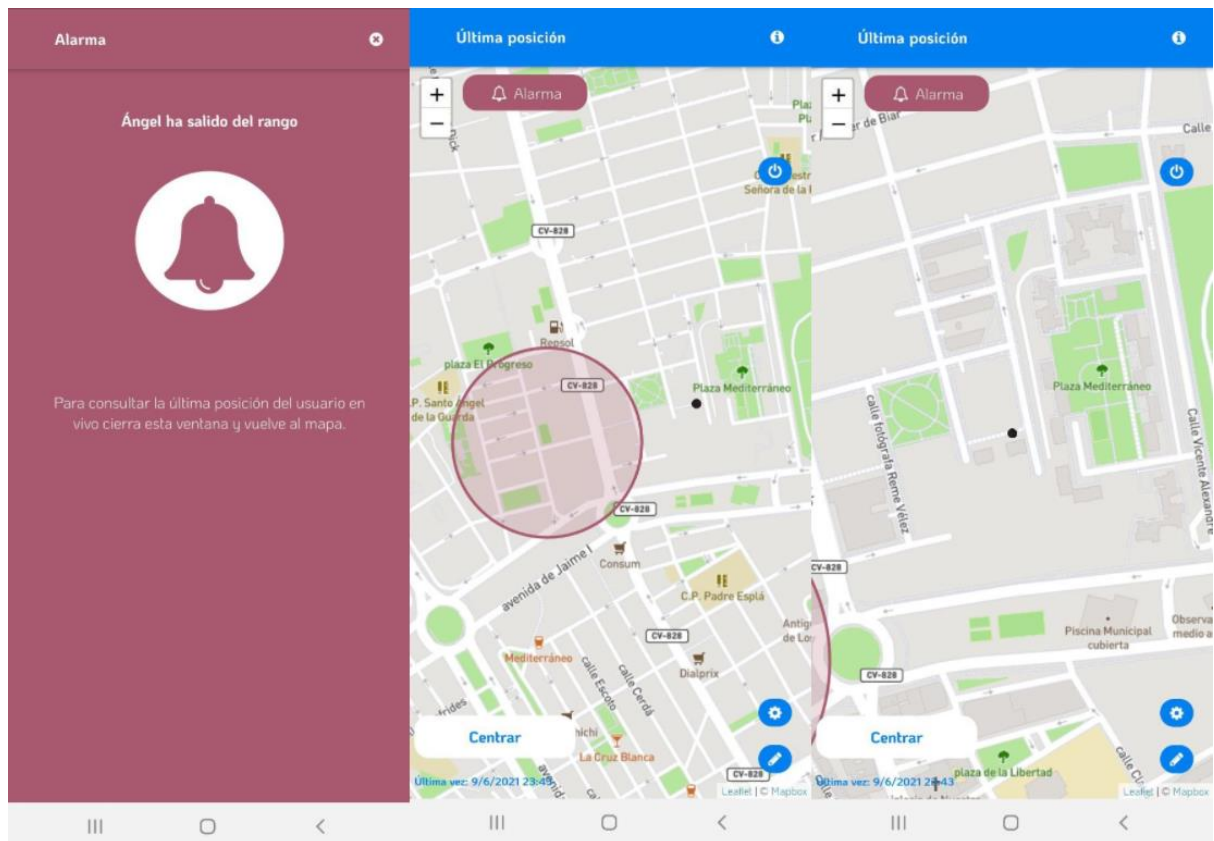


Figura 63. Interfaces de configuración del seguimiento para un usuario Guardián de la demostración (Fuente propia)

Como se puede ver en la figura anterior, se inserta el código de enlace que habría obtenido el usuario Protegido en la Figura 61, con el cual, si es correcto, se accederá a la página de seguimiento con el mapa.

La primera acción que se solicita al entrar nada más completar el enlace es la selección del rango de la zona segura del Protegido. Y, al confirmarla, se podrá observar esta en la pantalla de seguimiento. Llegado ese momento, solo queda esperar a que la alarma salte.



*Figura 64. Interfaces de alarma para un usuario Guardián de la demostración
(Fuente propia)*

Cuando el usuario Protegido salga de la zona seleccionada, la alarma saltará mostrando por primera vez su interfaz. Al cerrar esta, se regresa a la interfaz del mapa, en la que ahora se podrá ver un mensaje haciendo referencia a que la alarma ha saltado, un punto que marcará donde ha sido la última posición en la que se ha encontrado y cuándo ha sido recabada esa información.

Por último, en caso de no poder ver donde estuviese situado el punto que marca la posición actual del Protegido, presionando el botón centrar, el mapa se situará automáticamente donde el punto se encuentre.

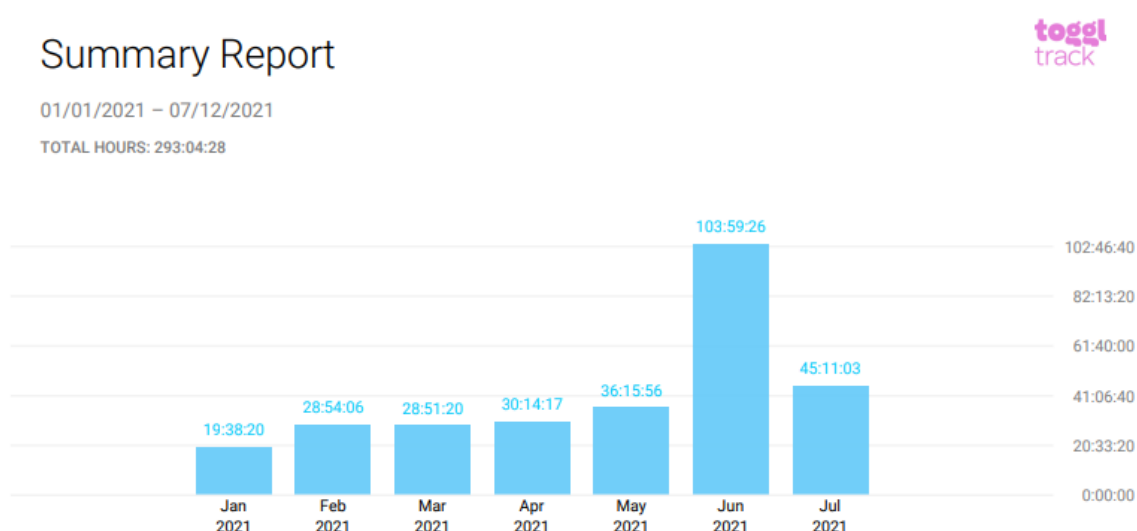
Llegado a este punto, se puede hacer balance del trabajo logrado respecto al producto final durante la implementación. Han sido implementadas el 100% de las interfaces diseñadas, aunque con cambios de diseño y con no todas las funcionalidades inicialmente planificadas, aunque si con un alto porcentaje de implementación de estas, que podría estimarse en un 80%, teniendo en cuenta que han faltado por implementar 6 de 30 inicialmente planificadas.

Estas son unas muy buenas cifras teniendo en cuenta la magnitud y dificultad de un proyecto de este calibre y que está finalizado.

11.2. Costes temporales

A la hora de contabilizar y analizar el tiempo invertido en el desarrollo del proyecto del Trabajo de Fin de Grado se ha hecho uso de la herramienta Toggl, como ya se menciona en el apartado de Metodología.

Gracias a esta herramienta se puede visualizar la cantidad de tiempo invertida por meses o semanas y por apartados de la memoria. En la siguiente figura se puede observar el tiempo invertido en el TFG durante los primeros meses del año, espacio de tiempo en el que se ha desarrollado todo el trabajo por completo.



*Figura 65. Horas invertidas en el TFG por mes
(Fuente propia)*

Se puede observar en el gráfico que, en los meses iniciales del desarrollo, se invirtió una cantidad de horas menor de la necesaria, motivo por el cual, en los meses cercanos a la entrega, se observa un aumento considerable de las horas dedicadas al proyecto.

Ese tramo de meses con un número menor de horas trabajadas en el TFG coincide además con el periodo de tiempo en el cual se tuvieron que cursar las prácticas curriculares de la carrera, hecho que implicó no poder invertir más tiempo en el proyecto.

Además, como se ha mencionado anteriormente, Toggl también muestra cuantas horas han sido invertidas en cada uno de los apartados de la memoria. En el gráfico de la siguiente figura se puede observar la relación de horas invertidas por apartados.



*Figura 66. Horas dedicadas a cada apartado de la memoria
(Fuente propia)*

Entre los apartados en los que se ha invertido más tiempo de trabajo se encuentran la Implementación, el Diseño y el Estado del Arte, entre los cuales destaca de forma evidente el apartado de Implementación. En la Implementación se ha invertido prácticamente el doble de horas que en el Diseño, siendo el segundo apartado con mayor tiempo invertido.

Como se puede observar en la Figura 65, el tiempo total invertido en el proyecto ha sido de 293h, por lo que se ha logrado prácticamente alcanzar las 300h a las que equivalen los 12 créditos del TFG, teniendo en cuenta que todavía falta por realizar el apartado posterior de Conclusiones y trabajo futuro.

Entre los tres apartados con más tiempo invertido suman 231h, tiempo que equivale al 79% de las horas invertidas totales en el proyecto. Este hecho indica la importancia que han tenido estos tres apartados en el proyecto, destacando la Implementación sobre el resto.

Haciendo una comparación de las horas invertidas con la planificación realizada de forma previa a implementar el proyecto, esta no se ha cumplido. En primer lugar, se planificó con el objetivo de defender en la convocatoria correspondiente a junio, hecho que no ha ocurrido puesto que se defenderá el proyecto en la convocatoria correspondiente a julio.

Tampoco se supo estimar de manera correcta el tiempo que conllevaría realizar las prácticas curriculares en empresa y el tiempo en semanas o meses que conllevaría cada apartado, puesto que dependiendo las horas invertidas en cada apartado este podía variar.

Aún con todo ello, el trabajo se ha finalizado para la convocatoria de julio, como también se mencionó como posibilidad en la Planificación del proyecto y el tiempo invertido es el correcto correspondiente a las 300h.

11.3. Asignaturas relacionadas

En este apartado se muestra la relación de este proyecto con asignaturas cursadas en la carrera de Ingeniería Multimedia y los beneficios que estas han aportado para una mejor realización del TFG.

De entre todas las asignaturas del grado, cabe destacar las cursadas en su cuarto y último curso, durante el cual se desarrolla el proyecto ABP, en este caso con el itinerario de Gestión de Contenidos.

Entre ellos, Sistemas de Difusión Multimedia, Sistemas Multimedia Avanzados y Proyectos Multimedia, en los cuales se aprende a gestionar y entender mejor la realización de proyectos de este calibre, ya sea en su implementación como en el desarrollo del documento de la memoria, sabiendo cuidar y justificar cada detalle o decisión tomada tratando de que sea de la mejor forma posible para el resultado final.

Además, otras asignaturas importantes en cuanto a conceptos y aprendizaje han sido Programación del Cliente Web y Desarrollo de Aplicaciones Web, en las cuales se empezaron a implementar aplicaciones de este estilo, Sistemas Multimedia y Análisis y Especificación de Sistemas Multimedia, aportando la capacidad de gestión y toma de decisiones en este tipo de proyectos, y Usabilidad y Accesibilidad, la cual aporta un punto de vista para poder diseñar un producto para un usuario que tratará hacer uso de él sin que este le genere ningún impedimento o dificultad.

12. Conclusiones y trabajo futuro

En este apartado, último del documento de la memoria del proyecto, se van a extraer conclusiones en referencia al desarrollo del proyecto, tanto en planificación, objetivos marcados, implementación como las impresiones personales sobre este.

12.1. Comprobación de objetivos

En primer lugar, el objetivo mayor en el proyecto era, precisamente, lograr realizar el proyecto, en este caso realizar una aplicación para notificar cuando alguien sale de una zona delimitada en un mapa, como dictan el título y subtítulo del documento.

Ese objetivo se ha logrado con creces, puesto que está logrado e incluso probado y, en este momento, se encuentra con estado funcional. Se ha implementado una aplicación teniendo en cuenta y siendo fiel a todas las decisiones tomadas en los apartados previos a la Implementación, dando como resultado una aplicación fiel a las decisiones tomadas y con todas las funcionalidades principales de esta implementadas.

Aunque en un inicio se esperaba poder acabar toda la aplicación de manera completa, el resultado obtenido con la aplicación final es positivo, pudiendo incluir funcionalidades secundarias, cuya no inclusión no implican malfuncionamiento de la app, como trabajo pendiente para el futuro.

A pesar de no haber cumplido el objetivo de publicar la aplicación en una tienda de aplicaciones como puede ser Google Play Store, debido a no estar finalizada esta de forma completa, sí se puede hablar de objetivos cumplidos y un balance positivo en la comprobación de objetivos.

12.2. Trabajo futuro y mejoras

Como ya se ha mencionado en varias ocasiones desde que se finalizó la implementación, el producto no incluye todas las funcionalidades inicialmente planteadas, por lo que en este apartado estas funcionalidades que serán implementadas en un futuro serán explicadas.

A corto plazo, las funcionalidades no implementadas pero destinadas a ser realizadas como trabajo futuro son:

- Posibilidad de realizar en enlace de seguimiento entre usuarios haciendo uso de un código QR. Con el objetivo de poder realizar de una forma más segura el enlace, puesto que los dispositivos de ambos usuarios deberán estar uno delante del otro para la lectura, implica que el código no puede ser copiado o traspasado entre más usuarios, al menos no de forma sencilla como un código de texto.
- La implementación del modo en segundo plano, haciendo uso del plugin nativo de Ionic Background Mode [56], el cual facilita el funcionamiento de la aplicación cuando esta no se encuentra abierta en pantalla en el dispositivo. Por motivos de tiempo no ha sido posible su inclusión, aunque el correcto funcionamiento de la aplicación genera buenas expectativas con su futura inclusión.
- Posibilidad de que un usuario Guardián pueda realizar acciones de seguimiento a más de un usuario Protegido. Aunque esta mejora está más enfocada a medio plazo, para el proyecto se definió una aplicación en la que un Guardián pudiese seguir únicamente a un solo usuario Protegido, siendo interesante la opción de valorar ampliarlo de cara al futuro.
- Que se puedan seleccionar más de una zona o rango de seguridad por seguimiento. Al igual que la mejora anterior, esta también sería una mejora a medio plazo y supondría un cambio respecto al planteamiento inicial de un único rango de seguridad por seguimiento. Esta mejora implicaría un mayor abanico de posibilidades para el seguimiento de un usuario Guardián.
- Finalizar la implementación de la interfaz de configuración. Esta, en su diseño, incluye la edición de todos los campos que son relativos a la información del usuario, como son nombre, email o contraseña, puesto que el tipo de usuario ya se puede cambiar. Por motivos de tiempo no ha sido posible implementar esos campos para su edición, pero se contempla poder realizarlo a corto plazo.
- Incluir sonido en la alarma y posibilidad de cambios entre los estados de sonido, vibración o silencio. En el diseño inicial de la interfaz mapa o de seguimiento se incluían opciones para realizar cambios entre diferentes estados. Aunque finalmente únicamente se implementó la opción de conectar o desconectar la alarma, se contempla como trabajo futuro incluir el resto de los estados como estaba inicialmente diseñado.

Además de estas mejoras, a largo plazo la aplicación podría ser ampliada todavía más, pudiendo almacenar un historial de posiciones para cuando un usuario Protegido haga saltar la alarma o incluso un botón de SOS para el Protegido que haga saltar la alarma se encuentre donde se encuentre, aunque estos son ejemplos que deberían ser estudiados y diseñados antes de su implementación.

Por último, de cara al futuro, el proyecto podría valorar su monetización. Como ya se mencionó en el Lean Canvas del Estudio de Viabilidad, el producto encajaría en un modelo Freemium, en el cual el usuario podría acceder a las funcionalidades principales sin coste alguno, pero debería pagar por otras mejoras secundarias. Mejoras secundarias como, por ejemplo, las futuras mejoras mencionadas como la posibilidad de establecer más de un rango de seguridad o capacidad de realizar el seguimiento a más de un usuario Protegido.

12.3. Impresiones personales

Realizar este trabajo ha sido un gran reto, puesto que en ninguna otra ocasión anteriormente había llevado a cabo un proyecto de esta magnitud. Todas las partes, desde la planificación, estudio de mercado y diseño hasta la implementación han supuesto un reto para hacer una buena toma de decisión y poder verlo reflejado en el resultado final, hecho que jamás había hecho en solitario sin ayuda o complementación de un equipo o grupo.

Debido a la motivación del proyecto, era un gran reto en el que había depositado una gran ilusión y ganas de hacer un buen trabajo.

En el apartado de objetivos se marcaron una serie de objetos personales a cumplir, entre los que se encontraban lograr empezar y acabar una aplicación funcional completa en la que he trabajado únicamente yo, poder crear y administrar una base de datos propia y que el resultado final de la aplicación fuese tangible y de buena calidad para poder ser parte de un portfolio personal.

Todos estos objetivos han sido cumplidos con creces, puesto que se ha logrado una aplicación completa y funcional, esta puede ser incluida como parte de un portfolio personal gracias a su gran calidad y la base de datos creada en Firebase, creada y gestionada por mí.

El resultado final e incluso este documento, con todas las explicaciones y justificaciones a cada decisión, representan un gran orgullo por un trabajo realizado con un gran respeto, con

muchas ganas de aprendizaje y poder ver resultados y con la sensación de que el resultado de todo el trabajo y tiempo invertidos ha sido óptimo.

Durante el desarrollo del trabajo se ha aprendido a gestionar el tiempo de una forma más útil y productiva, se ha ganado en experiencia para futuros proyectos similares, ya sea experiencia tanto en la implementación como en la gestión del proyecto, se han podido poner en práctica conocimientos, que hasta el momento eran únicamente teóricos, y mejorarlos.

Este trabajo ha sido un gran reto para mejorar y aprender, por el cual estoy muy satisfecho con el resultado, las decisiones tomadas y el buen trabajo realizado.

El resultado final puede ser interesante para futuros usuarios o clientes ajenos al desarrollo del proyecto, que podrían ver la aplicación como solución a problemas reales actuales. Aunque todavía en desarrollo, al finalizar su implementación por completo en el futuro, será interesante publicarla, conocer su acogida y la cantidad de uso que pueda recibir.

Referencias

1. *Análisis DAFO. Concepto de Análisis DAFO.* *elEconomista.es* Diccionario de economía. Disponible en: <https://www.eleconomista.es/diccionario-de-economia/analisis-dafo>
2. CECARM. Consultas y FAQs. ¿Qué es el Lean Canvas? Disponible en: <https://www.cecarm.com/empreendedor/estrategia/consultas-y-faqs/que-es-el-lean-canvas-3801>
3. EL PAÍS. SMART LIFE. Cómo publicar una app en Google Play y cuánto cuesta. Disponible en: https://cincodias.elpais.com/cincodias/2015/02/01/lifestyle/1422792260_243066.html#:~:text=Por%20lo%20tanto%2C%20el%20precio,tengan%20disponibles%20en%20la%20tienda
4. Apple. Developer. Compra y activación. Precios. Disponible en: <https://developer.apple.com/es/support/purchase-activation/#:~:text=Precios,durante%20el%20proceso%20de%20inscripci%C3%B3n>
5. EVALUANDO SOFTWARE.COM. Qué es la geolocalización y cómo funciona. Disponible en: <https://www.evaluandosoftware.com/la-geolocalizacion-funciona/>
6. Sistemas de posicionamiento móvil para interiores vía WIFI. TFG Jesús Piedra Vega, Ingeniería Técnica de Informática de sistemas, junio de 2016. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/53021/7/pfi45js390pTFC0716mem%C3%B2ria.pdf>
7. Gabinete Jurídico-Tecnológico Juande Meseguer. Especialistas en Derechos fundamentales y Nuevas Tecnologías. Localización y posicionamiento de móviles mediante antenas (BST y CGI) y GPS. 4.- Problemática. Disponible en: <https://gabinetejuridicotecnologicojuandemeseguer.es/localizacion-posicionamiento-moviles-mediante-sistemas-moviles-gps#:~:text=La%20localizaci%C3%B3n%20GSM%20o%20G,f%C3%ADsicamente%20un%20terminal%20m%C3%B3vil%20determinado>
8. ¿Qué tipos de localizadores GPS existen? mapit. Disponible en: <https://mapit.me/2017/01/24/tipos-localizadores-gps/>
9. Anillamiento. ¿Qué es el anillamiento? Grupo Ibérico de Anillamiento. Disponible en: <http://www.gia-anillamiento.org/anillamiento.asp>
10. Técnicas empleadas. El Proyecto. SEO/BirdLife. Sociedad Española de Ornitología. Disponible en: <http://www.migraciondeaves.org/tecnicas.php>

11. LoPe. Localizador de personas. Teleasistencia. Cruz Roja. Disponible en: <https://www.cruzroja.es/principal/web/teleasistencia/localizador-de-personas>
12. Weenect. Localizadores GPS para la familia. Disponible en: <https://www.weenect.com/es/>
13. Weenect – GPS. Google Play. Disponible en: <https://play.google.com/store/apps/details?id=com.maptitebalise.MPB&gl=ES>
14. Find My Kids: localiza niños con móvil y reloj GPS. Google Play. Disponible en: https://app.adjust.com/get4elc?campaign=findmykids_org&adgroup=ru&creative=main-page
15. Pingo: chat with parents. Chatea con tus padres. Google Play. Disponible en: <https://play.google.com/store/apps/details?id=org.findmykids.child>
16. Escucha lo que ocurre alrededor de tu hijo. La app <<Find My Kids>> te ayudará a dejar de preocuparte por la seguridad de tu hijo. Findmykids. Disponible en: <https://findmykids.org/es>
17. Life360 – Localizador Familiar y Móvil. Google Play. Disponible en: <https://market.android.com/details?id=com.life360.android.safetymapd>
18. GeoZilla GPS Localizador Familiar. Google Play. Disponible en: <https://play.google.com/store/apps/details?id=com.geozilla.family>
19. GeoZilla – Localizador GPS. Apple Store. Disponible en: <https://apps.apple.com/mx/app/geozilla-localizador-gps/id981856216?platform=iphone>
20. AlertCops. Google Play. Disponible en: <https://play.google.com/store/apps/details?id=com.alertcops4.app&hl=es>
21. AlertCops. La app de la Policía y la Guardia Civil. Ministerio del Interior. Gobierno de España. Disponible en: <https://alertcops.ses.mir.es/mialertcops/>
22. ¿Qué es un objetivo SMART?. Para gente inteligente, objetivos Smart. AQ Ana Caldas. Ana Caldas González. Disponible en: <https://aqaldas.com/para-gente-inteligente-objetivos-smart/>
23. Qué es Kanban: Definición, Características y Ventajas. kanbanize. Disponible en: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>
24. Scrum: qué es y cómo funciona esta metodología. Global Growth Agents. Wam. Disponible en: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>
25. IEEE. (2008). Especificación de Requisitos según el estándar de IEEE 830. Disponible en: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

26. Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar. acens the Cloud services company, una compañía de Telefónica. Disponible en: <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
27. Introducción. Cloud Firestore. Documentación. Firebase. Disponible en: <https://firebase.google.com/docs/firestore?hl=es>
28. Cuotas y límites. Cloud Firestore. Recursos. Google Cloud. Disponible en: <https://cloud.google.com/firestore/quotas?hl=es>
29. Comienza a usar las reglas de seguridad de Cloud Firestore. Proteger y validar datos. Cloud Firestore. Documentación. Firebase. Disponible en: <https://firebase.google.com/docs/firestore/security/get-started?hl=es>
30. Estructura reglas de seguridad de Cloud Firestore. Proteger y validar datos. Cloud Firestore. Documentación. Firebase. Disponible en: <https://firebase.google.com/docs/firestore/security/rules-structure?hl=es>
31. El Mapa de Experiencia del Cliente o Customer Journey Map. Alfronso Prim. innokabi. Disponible en: <https://innokabi.com/claves-para-emocionar-a-tu-cliente-customer-journey-map/>
32. KoHo. Google Fonts. Disponible en: <https://fonts.google.com/specimen/KoHo>
33. Figma. Disponible en: <https://www.figma.com/>
34. Firebase Crashlytics. Products. Firebase. Disponible en: <https://firebase.google.com/products/crashlytics>
35. Firebase Performance Monitoring. Products. Firebase. Disponible en: <https://firebase.google.com/products/performance>
36. Firebase Test Lab. Products. Firebase. Disponible en: <https://firebase.google.com/products/test-lab>
37. Code editing. Redefined. Visual Studio Code. Disponible en: <https://code.visualstudio.com/>
38. Where the world builds software. Github. Disponible en: <https://github.com/>
39. One codebase. Any platform. Ionic framework. Disponible en: <https://ionicframework.com/>
40. One framework. Mobile & desktop. Angular. Disponible en: <https://angular.io/>
41. Typed JavaScript at Any Scale. TypeScript. Disponible en: <https://www.typescriptlang.org/>
42. Chrome. Google. Disponible en: <https://www.google.com/intl/es/chrome/>
43. The oficial Angular library for Firebase. Github. AngularFire. Disponible en: <https://github.com/angular/angularfire>

44. *Maps and Location for developers.* mapbox. Disponible en: <https://www.mapbox.com/>
45. *Apache Cordova.* Disponible en: <https://cordova.apache.org/>
46. *Build fast, responsive sites with Bootstrap.* Bootstrap. Disponible en: <https://getbootstrap.com/>
47. *Get vector icons and social logos on your website with Font Awesome, the web's most popular icon set and toolkit.* Font Awesome. Disponible en: <https://fontawesome.com/>
48. *Icon Fonts generator.* Fontello. Disponible en: <https://fontello.com/>
49. *NavController. Navigation. API. Documentation.* Ionic Framework. Disponible en: <https://ionicframework.com/docs/v3/api/navigation/NavController/>
50. *Maps and Location for developers.* Mapbox. Disponible en: <https://www.mapbox.com/>
51. *An open-source JavaScript library for mobile-friendly interactive maps.* Leaflet. Disponible en: <https://leafletjs.com/>
52. *Geolocation. Plugins. Native. Documentation.* Ionic Framework. Disponible en: <https://ionicframework.com/docs/native/geolocation>
53. *Java SE Development Kit 8 Downloads.* Java SE. Technical Details. Java. Oracle. Disponible en: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
54. *Android Studio provides the fastest tools for building apps on every type of Android device.* Android Studio. Developers. Android. Disponible en: <https://developer.android.com/studio>
55. *Accelerate Developer productivity.* Gradle Build Tool. Disponible en: <https://gradle.org/>
56. *Background Mode. Plugins. Native. Documentation.* Ionic Framework. Disponible en: <https://ionicframework.com/docs/native/background-mode>